

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re PATENT APPLICATION of : ATTN: OFFICE OF PUBLICATIONS
Kyoung-Shik JUN et al. : Attorney Docket No.: SEC.911
Serial No.: (NEW) :
Filed: 12 February 2002 :
For: "METHOD FOR CONTROLLING A PROCESSING APPARATUS"

1017 U.S. PRO
10/072901



#2
Dronh
apee
Rutsee
5/8/02

CLAIM OF PRIORITY

Honorable Assistant Commissioner for Patents and Trademarks,
Washington, D.C. 20231

Sir:

Applicants, in the above-identified application, hereby claim the priority date under the
International Convention of the following Korean application:

Appln. No.: 2001-0039371 filed: 3 July 2001


as acknowledged in the Declaration of the subject application.

A certified copy of said application is being submitted herewith.

Respectfully submitted,

VOLENTINE FRANCOS, P.L.L.C.

Date: 12 February 2002

By: 
Kenneth D. Springer
Registration No. 39,843

VOLENTINE FRANCOS, P.L.L.C.
12200 Sunrise Valley Drive, Suite 150
Reston, Virginia 20191
Tel. (703) 715-0870
Fax. (703) 715-0877

대한민국 특허청

KOREAN INTELLECTUAL PROPERTY OFFICE

11017 U.S. PTO
10/072901
02/12/02

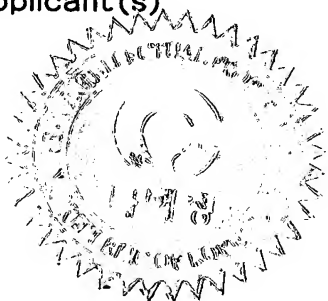
별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.

출원번호 : 특허출원 2001년 제 39371 호
Application Number PATENT-2001-0039371

출원년월일 : 2001년 07월 03일
Date of Application JUL 03, 2001

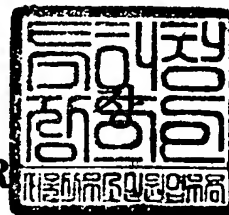
출원인 : 삼성전자 주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2001 07 21
 년 월 일

특 허 청

COMMISSIONER



【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【제출일자】	2001.07.03
【발명의 명칭】	공정장치의 제어방법
【발명의 영문명칭】	METHOD FOR CONTROLLING IN PROCESSING APPARATUS
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	박영우
【대리인코드】	9-1998-000230-2
【포괄위임등록번호】	1999-030203-7
【발명자】	
【성명의 국문표기】	전경식
【성명의 영문표기】	JUN,Kyoung Shik
【주민등록번호】	630804-1030119
【우편번호】	463-821
【주소】	경기도 성남시 분당구 서현동 87번지 한신아파트 112-190
【국적】	KR
【발명자】	
【성명의 국문표기】	박찬훈
【성명의 영문표기】	PARK,Chan Hoon
【주민등록번호】	620418-1448822
【우편번호】	143-210
【주소】	서울특별시 광진구 광장동 현대아파트 502동 1803호
【국적】	KR
【발명자】	
【성명의 국문표기】	박일석
【성명의 영문표기】	PARK,Yil Seug
【주민등록번호】	650122-1531818

【우편번호】	441-390
【주소】	경기도 수원시 권선구 권선동 현대아파트 202-1103
【국적】	KR
【발명자】	
【성명의 국문표기】	조봉수
【성명의 영문표기】	CHO, Bong Su
【주민등록번호】	661217-1026227
【우편번호】	463-905
【주소】	경기도 성남시 분당구 이매동(이매촌) 청구아파트 611-40
【국적】	KR
【발명자】	
【성명의 국문표기】	강현태
【성명의 영문표기】	KANG, Hyun Tai
【주민등록번호】	680306-1235115
【우편번호】	449-902
【주소】	경기도 용인시 기흥읍 공세리 382-1 호수청구아파트 105-201
【국적】	KR
【발명자】	
【성명의 국문표기】	제영호
【성명의 영문표기】	JEI, Young Ho
【주민등록번호】	740208-1914713
【우편번호】	604-851
【주소】	부산광역시 사하구 하단2동 496-23 (25/1) 영진주택 308
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인 박영우 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	55 면 55,000 원

【우선권주장료】	0	건	0	원
【심사청구료】	18	항	685,000	원
【합계】	769,000	원		
【첨부서류】	1.	요약서·명세서(도면)_1통		

【요약서】

【요약】

본 발명은 공정장치의 제어방법에 관한 것으로서, 특히 피처리물을 처리하기 위한 공정장치의 입력값과, 상기 공정장치에서 처리된 피처리물을 측정하여 얻은 측정값 사이의 오차값을 구하고, 상기 오차값을 줄이는 방향으로 상기 공정장치의 입력값을 보정하기 위한 보정값을 산출하고, 상기 값들을 다음 보정값 산출시에 사용하기 위한 공정 데이터로 관리한다. 상기 공정장치에 로딩되는 피처리물과 동일 히스토리를 갖는 이전 공정 데이터를 검색하고, 상기 검색된 동일 히스토리를 갖는 이전 공정 데이터들 중 최근 복수개의 이전 보정값들로부터 현재 바이어스 보정값을 추정하고, 이전 공정 데이터들 중 가장 최근 복수개의 이전 랜드 보정값들을 기반으로 신경회로망에 의해 현재 랜드 보정값을 추정한다. 상기 공정장치의 현재 보정값으로 상기 추정된 바이어스 보정값과 랜드 보정값을 합산한다. 상기 오차값을 사용하여 상기 랜드 보정값의 변화를 추종하도록 상기 신경회로망을 학습시킨다. 따라서, 본 발명에서는 보정값을 이전 히스토리와의 연관성을 갖는 바이어스 성분과 임의적 성분인 랜드성분으로 분리하고, 바이어스성분은 히스토리를 바탕으로 추정하고, 랜드성분은 신경회로망을 이용하여 오차를 최소화함으로써 데이터 사용기기한의 제약조건을 완화시킬 수 있어서 샘플링 공정의 횟수를 획기적으로 감소시킬 수 있다.

【대표도】

도 1

【명세서】

【발명의 명칭】

공정장치의 제어방법 {METHOD FOR CONTROLLING IN PROCESSING APPARATUS}

【도면의 간단한 설명】

도 1은 본 발명에 의한 노광장치의 제어시스템의 블록도.

도 2는 오버레이 측정기의 측정 파라미터를 설명하기 위한 도면.

도 3은 노광장치의 대략적인 구성을 나타낸 도면.

도 4는 오버레이 측정기로부터 오버레이 보정제어기로 다운되는 측정 데이터의 구조를 나타낸 도면.

도 5는 오버레이 보정제어기로부터 스텝퍼에 입력되는 보정 데이터의 구조를 나타낸 도면.

도 6은 시간에 따른 스텝퍼의 보정값 변동을 나타낸 그래프.

도 7은 도 6의 보정값들을 동일 히스토리 별로 재배열한 상태를 나타낸 그래프.

도 8은 도 7의 보정값에서 동일 히스토리의 평균값(바이어스값)을 제거한 상태를 나타낸 그래프.

도 9는 스텝퍼의 OFFSET-X에 대한 보정값의 시간에 따른 변동상황을 나타낸 그래프

도 10은 도 9의 보정값의 주파스 스펙트럼을 나타낸 그래프.

도 11은 본 발명에 의한 신경회로망의 일 예를 나타낸 도면.

도 12 내지 도 20은 본 발명에 의한 오버레이 보정 제어기의 보정값 추정 동작을 설명하기 위한 플로차트.

도 21 내지 도 25는 본 발명에 의한 오버레이 보정 제어기와 기존 제어기의 측정오차 분포를 비교한 그래프.

도 26 내지 도 35는 본 발명에 의한 오버레이 보정제어 알고리즘을 실제 장비에 적용한 결과를 분석한 그래프.

*도면의 주요 부분에 대한 부호의 설명

10 : 포토장치 12 : 코팅처리부

14 : 얼라인 및 노광처리부 16 : 현상처리부

20 : 오버레이 측정기 30 : 오버레이 보정 제어기

40 : 노광시스템 42 : 웨이퍼 스테이션

44 : 렌즈 46 : 레티클

50 ; 입력층 52 : 제 1 은닉층

54 ; 제 2 은닉층 56 : 제 2 은닉층

58 : 출력층 W : 웨이퍼

MS : 미스 얼라인먼트 사이트 OM : 외측 얼라인먼트 마크

IM ; 내측 얼라인먼트 마크

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

- <26> 본 발명은 공정장치의 제어방법에 관한 것으로서, 특히 반도체 소자 제조공정 중 노광공정의 스텝퍼의 보정값을 추정하기 위한 새로운 에이엘엠에스-엔엔(ALMS-NN : Adaptive Least Mean Square Neural Network) 알고리즘에 의한 공정장치의 제어방법에 관한 것이다.
- <27> 반도체 산업에서 경쟁력 강화를 위한 일환으로 높은 생산 수율을 보장할 수 있는 효율적인 생산 시스템을 구축하기 위해 수많은 연구를 진행하고 있다. 특히 핵심 반도체 제조 공정들 중의 하나인 노광 공정(Photo-lithographic Process)의 경우에도 공정 조건의 변화가 빈번하여 이에 대처할 수 있는 체계적인 생산 시스템 개발이 필요한 실정으로, 생산 수율을 높이기 위해서 샘플링(Sampling) 공정의 빈도를 줄이기 위한 시스템 구축을 위해 많은 노력을 하고 있다.
- <28> 이러한 시스템 구축을 위해 노광 공정에서 우선적으로 고려해야 할 사항으로 공정상의 정열오차 문제가 있으며, 이것은 공정의 물리적, 화학적 특성 해석의 어려움과 공정 중에 개입되는 잡음 및 공정 후의 측정 오차에 의해 주로 발생하는 것으로 생산 수율에 직접적으로 영향을 주는 샘플링의 빈도를 증가시키는 원인이 된다.
- <29> 반도체 제조 공정에서 높은 생산 수율을 보장할 수 있는 생산 시스템으로 많이 개발, 사용하고 있는 공정 제어 시스템(Process Control System; PCS)은 공정 진행과정에 대한 수학적 모델이 존재하지 않고, 과거에 진행된 공정 데이터를 가공하여 통계적 수치

에 의해서 공정을 제어하는 경우가 대부분이었다.

<30> 일종의 ‘경험 전수’ 방식으로 이전 공정들의 경험치를 현 공정에 반영시키는 방식은 최근 수행한 동일 히스토리 데이터(History Data)에 대하여 가중치 평균을 피드백시키는 알고리즘이다. 그러나, 이러한 알고리즘은 시스템의 시변성 특성을 고려하지 않는 정적(static) 알고리즘이므로 정해진 기간 내에 동일 히스토리 데이터가 부족하거나 또는 연속된 스펙 아웃(spec-out)의 발생으로 빈번하게 샘플링 공정을 수행하여야 하는 문제점이 있다.

<31> 이에, 특정한 수학적 모델이 존재하지 않고, 비선형적인 시스템으로 이루어진 공정에 적절하게 대처하기 위한 방법으로 최근에 신경회로망 모델을 이용한 공정 제어기 설계기법들(S. Limanond, J. Si, and K. Tsakalis, “Monitoring and Control of Semiconductor Manufacturing Processes”, IEEE Control Systems, 1998. /X. A. Wang and R. L. Mahajan, “Artificial Neural Network Model-Based Run-to-Run Process Controller”, IEEE Trans. on Component, Packaging, and Manufacturing Technology-Part C, vol. 19, no. 1, Jan. 1996.)이 제시되고 있다.

<32> 첫째, 과거에 진행된 공정 데이터를 이용하여 신경회로망을 학습시켜 이전 공정에 대해서 패턴탐색을 통해 제조 공정을 예측하는 방법이 있다. 이 방법의 기본 가정은 비선형적인 시스템의 변화패턴이 완전하게 임의적이지는 않을 것이라는 것에 염두해 두고, 하나의 스텝퍼(stepper) 장치에 대한 시스템의 변화패턴이 완전히 임의적인 것이 아니라면 최근의 변화패턴과 유사한 패턴을 가지는 과거의 history 데이터가 존재할 것이라 예상할 수 있으므로 이 과거 시스템의 변화패턴을 이용하여 현재의 출력값을 추정하는 것이다.

<33> 그러나, 이러한 방법은 효과적인 적용을 위해서 많은 양의 과거 히스토리 데이터를 필요로 하게 되어 데이터 운영상의 어려움이 있을 수 있고 매번 반복되는 패턴 탐색과 신경회로망 학습으로 인해 계산량이 늘어난다는 단점을 가지게 된다.

<34> 둘째, 많이 사용되고 있는 EWMA(Exponential Weighted Moving Average) 방식은 타임시리즈(Time Series) 방식으로 기술될 수 있는 데이터 변화를 가진 시스템의 모델링 및 예측 방법을 제공하는 데, 특히 반도체 제조의 공정 제어분야에서 널리 사용되고 있는 방식들로 다음 문헌들(E. Sachs, A. Hu, and A. Ingolfsson, "Run by Run Process Control: Combining SPC and Feedback Control", IEEE Trans. on Semiconductor Manufacturing, vol. 8, no. 1, Feb. 1995./N. S. Patel and S. T. Jenkins, "Adaptive Optimization of Run-to-Run Controllers: The EWMA Example", IEEE Trans. on Semiconductor Manufacturing, vol. 13, no. 1, Feb. 2000./T. H. Smith, D. S. Boning, "A Self-Tuning EWMA Controller Utilizing Artificial Neural Network Function Approximation Techniques", IEEE Trans. on Components, Packaging, and Manufacturing Technology-Part C, vol. 20, no. 2, April, 1997.)에 개시되어 있다.

<35> EWMA 방법은 모델이 간단할 뿐만 아니라 다음 수식과 같이 간단한 회귀적(recursive) 형태의 적용이 가능하기 때문에 실제 반도체 장치 운영에 많이 이용되고 있는 방식이다.

<36>
$$\hat{x}(i) = \lambda x(i-6) + (1-\lambda) \hat{x}(i-1)$$

<37> 그러나, EWMA 방식을 적용할 때 작은 λ 값을 사용하는 경우에는 과거의 데이터들도 무시할 수 없는 웨이트가 적용되기 때문에 올바른 평가(estimation)를 위해서는 많은

과거 데이터가 필요하게 된다는 단점이 있다.

<38> 세째, 칼만 필터링(Kalman Filtering)을 이용한 시스템 예측 기법은 기본적인 시스템의 동특성이 상태공간(state-space) 형태의 미분방정식(Differential Equation) 또는 차등방정식(Difference Equation)으로 모델링되며 백색 잡음(White Noise)에 의해 간섭받고 있는 시스템에 대한 고전적인 예측기법이다.

<39> 시스템의 보정값이 노이즈에 의해 간섭받지 않을 때는 보정값의 변화가 없지만 노이즈에 의해 일반적으로 그 값이 변화한다는 점에 근거하여 보정값의 변화특성을 다음과 같은 선형화된 모델로 가정한다.

<40>
$$x(k+1) = x(k) + w(k)$$

<41> 여기서, $w(k)$ 는 보정값의 변화의 원인이 되는 화이트 노이즈 항목이다.

<42> 이 때, 가정한 시스템 모델이 원래 시스템의 동특성 및 잡음 특성을 얼마나 잘 묘사하고 있는가에 따라 성능이 결정되는데, 반도체 공정과 같이 비선형적 특성이 강한 시스템에서 가정한 모델을 원래 시스템에 가깝게 모델링하기란 쉽지 않다.

【발명이 이루고자 하는 기술적 과제】

<43> 본 발명의 목적은 이와 같은 종래 기술의 문제점을 해결하기 위하여 많은 양의 과거 히스토리 데이터에 의존하지 않으면서 잦은 피가공물의 교체로 인해 샘플링 의존도가 높은 공정에 대해서 효과적으로 적용할 수 있는 새로운 ALMS-NN 알고리즘을 가진 공정 장치의 제어방법을 제공하는 데 있다.

<44> 본 발명의 다른 목적은 노광 공정의 스텝퍼 장치를 대상으로 중첩 정렬 오차

(Overlay Alignment Error)를 효과적으로 보정해 주는 장치 입력값을 결정하기 위한 새로운 ALMS-NN 알고리즘을 가진 노광장치의 제어방법을 제공하는 데 있다.

【발명의 구성 및 작용】

<45> 상기 목적을 달성하기 위한 본 발명의 제어방법은 피처리물을 처리하기 위한 공정 장치의 입력값과, 상기 공정장치에서 처리된 피처리물을 측정하여 얻은 측정값 사이의 오차값을 구하고, 상기 오차값을 줄이는 방향으로 상기 공정장치의 입력값을 보정하기 위한 보정값을 산출하고, 상기 값들을 다음 보정값 산출시에 사용하기 위한 공정 데이터로 관리한다. 상기 공정장치에 로딩되는 피처리물과 동일 히스토리를 갖는 이전 공정 데이터를 검색하고, 상기 검색된 동일 히스토리를 갖는 이전 공정 데이터들 중 최근 복수개의 이전 보정값들로부터 현재 바이어스 보정값을 추정하고, 이전 공정 데이터들 중 가장 최근 복수개의 이전 랜드 보정값들을 기반으로 신경회로망에 의해 현재 랜드 보정값을 추정한다. 상기 공정장치의 현재 보정값으로 상기 추정된 바이어스 보정값과 랜드 보정값을 합산하고, 상기 오차값을 사용하여 상기 랜드 보정값의 변화를 추종하도록 상기 신경회로망을 학습시킨다.

<46> 즉, 본 발명에서는 보정값 $x(n)$ 의 효과적 추정을 위해 $x(n)$ 을 히스토리에 연관성이 있는 바이어스 성분($x_{bias}(n)$)과 변화원인을 정확히 알 수 없는 임의적 성분($x_{rand}(n)$)으로 분리하고, $x_{bias}(n)$ 는 해당 롯트(Lot)의 히스토리를 바탕으로 추정하고, $x(n)$ 의 정확한 추정을 힘들게 하는 요소인 $x_{rand}(n)$ 성분은 그 임의적 성질 때문에 정확한 추정은 거의 불가능하므로 신경회로망의 역전파 학습을 이용하여 추종(tracking)하게 함으로써 $x(n)$ 의 추정오차를 최소화 한다.

<47> 이때, $x_{rand}(n)$ 성분은 해당 룯트의 히스토리에 대한 연관성이 제거된 상태가 되기 때문에 해당 룯트의 히스토리에 상관없이 모든 데이터를 이용할 수가 있다.

<48> 그러므로, 본 발명에 의해 제안된 알고리즘에 의해 계산되는 $x(n)$ 는 히스토리에 따른 ($x_{bias}(n)$)의 시간적 변화가 크지 않을 경우 기존 시스템에서 문제가 되었던 데이터 사용기한의 제약조건이 해결되고, 해당 룯트의 히스토리 이외에 $x(n)$ 에 영향을 미치는 외부적 요인이 변화할 경우에도 신경회로망이 지닌 학습능력을 이용하여 효과적으로 대처할 수 있게 된다.

<49> 따라서, 소자 변경이 많이 일어나는 생산라인에서도 과거 히스토리 데이터에 의존하는 경우가 줄어들어 샘플링 공정의 회수를 획기적으로 줄일 수 있다.

<50> 본 발명에서 현재 바이어스 보정값 추정단계는 다음 수식으로 정의된 구간선형 가중평균 알고리즘에 의해 추정한다.

$$\text{<51> } x_{bias} = \frac{1}{W} \sum_{i=n-W+1}^n \left[\frac{W+n-i}{\sum_{j=1}^W i} x^{sh}(i) \right]$$

<52> 여기서, x_{bias} 는 바이어스 보정값, W 는 구간, x^{sh} 는 동일 히스토리를 갖는 이전 바이어스 보정값을 나타낸다.

<53> 본 발명에서 현재 랜드 보정값은 다층 신경망을 통하여 오류 역전파 학습법에 의해 랜드 보정값의 오차를 줄이는 방향으로 추종함으로써 획득한다.

<54> 본 발명의 다른 목적을 달성하기 위하여 본 발명의 노광장치는 웨이퍼 상의 포토레지스트를 노광하기 위한 노광장치의 입력값과, 상기 노광장치에서 노광처리되어 현상된

포토레지스트 패턴을 오버레이 측정장치를 통하여 측정하여 얻은 측정값과의 오차값을 구하고, 상기 오차값을 줄이는 방향으로 상기 입력값을 보정하기 위한 보정값을 산출하고, 상기 값들을 다음 보정값 산출시에 사용하기 위한 노광공정 데이터를 생성시간단위로 관리한다. 상기 노광장치에 로딩되는 새로운 릿트와 동일 히스토리를 갖는 이전 공정 데이터를 검색하고, 상기 검색된 동일 히스토리를 갖는 이전 공정 데이터들 중 최근 복수개의 이전 보정값들로부터 현재 보정값의 바이어스 성분을 추정하고, 이전 공정 데이터들 중 가장 최근 복수개의 이전 랜드 보정값들을 기반으로 신경회로망에 의해 현재 보정값의 랜드 성분을 추정한다. 상기 노광장치의 현재 보정값으로 상기 추정된 바이어스 성분과 랜드 성분을 합산하고, 상기 오차값을 사용하여 상기 랜드 성분의 변화를 추종하도록 상기 신경회로망을 학습시킨다.

<55> 본 발명에서 검색단계에서 히스토리 구성요소인 레티클, PPID(PROCESS PROGRAM ID), 베이스 I, 베이스 II가 모두 동일한 데이터를 동일 히스토리 공정 데이터로 검출한다.

<56> 또한, 검색단계에서 동일 히스토리의 공정 데이터가 존재하지 않을 경우에는 상기 레티클 요소가 동일한 공정 데이터들 중에서 나머지 요소들의 우선순위에 따라 보정값의 바이어스 부분을 유추한다.

<57> 본 발명의 유추방법은 히스토리 구성요소 중 어느 하나의 구성요소만 다른 공정 데이터들을 추출하고, 상기 추출된 하나의 구성요소만 다른 공정 데이터들 중 어느 하나의 구성요소의 상대값을 이용하여 보정값의 바이어스 성분을 유추하고, 상기 상대값을 이용하여 바이어스 성분을 산출할 수 없을 경우에는 상기 추출된 어느 하나의 구성요소만 다른 공정 데이터들의 평균값을 구하여 보정값의 바이어스 성분으로 유추한다.

- <58> 이하 첨부한 도면을 참조하여 본 발명을 보다 상세히 설명하고자 한다.
- <59> 도 1은 본 발명에 의한 노광장치의 제어시스템의 블록구성을 나타낸다.
- <60> 포토장비(10)는 코팅처리부(12), 얼라인 및 노광처리부(14) 및 현상처리부(20)를 포함한다.
- <61> 포토장비(10)는 웨이퍼(W) 상에 피식각층을 데포(deposition)하고, 피식각층 상에 포토레지스트 패턴을 형성하고, 포토레지스트 패턴을 식각 마스크로 사용하여 피식각층을 식각하는 포토리소그래피(photo lithography) 공정에 의해 웨이퍼 상에 하나의 패턴층을 형성한다. 이와 같은 공정을 각층마다 반복하여 다층의 패턴층들을 중첩 형성하여 원하는 회로패턴을 웨이퍼 상에 형성함으로써 하나의 웨이퍼 상에 다수의 집적회로칩들을 만든다.
- <62> 그러므로, 포토리소그래피 공정은 반도체 소자의 제조공정에 있어서 생산수율에 지대한 영향을 미치는 매우 중요한 핵심 공정이라 할 수 있다.
- <63> 포토공정은 크게 코팅공정, 얼라인먼트 및 노광공정, 현상공정으로 구분될 수 있다.
- <64> 코팅처리부(12)에서는 웨이퍼 표면의 습기를 제거하고, 도포될 포토레지스트와 웨이퍼 표면과의 밀착성을 증가시키기 위하여 프리 베이크 공정과 고압 순수과 브러쉬를 이용하여 웨이퍼 표면의 불순물을 제거하는 스크러빙공정, 균일한 코팅을 위한 스핀공정, 솔벤트를 휘발시키고 포토레지스트를 경화시키는 소프트 베이크 공정 등을 수행한다.
- <65> 얼라인 및 노광처리부(14)에서는 스탭퍼의 기준마크에 의해 레티클을 정렬시키고, 웨이퍼와 레티클을 정렬시키는 프리 얼라인공정, 웨이퍼의 플랫존을 고정시키는 얼라인

공정, 노출량을 결정하여 포토레지스트를 노광시키는 노광공정 등을 수행한다.

<66> 현상처리부(16)에서는 정재과 효과를 제거하는 포스트 노광공정, UV광과 반응한 부분을 선택적으로 제거하는 현상공정, 웨이퍼에 남겨진 포토레지스트 패턴이 충분히 열적 환경에 견딜 수 있도록 경화시키는 하드 베이킹 공정 등을 수행한다.

<67> 이와 같이 포토장비(10)를 통하여 웨이퍼(W) 상에 포토레지스트 패턴을 형성시킨 다음에 하부 패턴층과의 중첩된 위치의 미스얼라인먼트를 측정하여 오차허용한계 이내의 값을 가지는 가를 오버레이 측정기(20)를 통하여 반드시 측정을 통해 확인해야 한다.

<68> 오버레이 측정기(20)에서는 중첩된 패턴층들의 모든 패턴들을 모두 비교하여 미스얼라인먼트를 측정할 수는 없다. 따라서, 웨이퍼(W) 상에 부가된 미스얼라인먼트 측정 사이트를 통하여 미스얼라인먼트를 측정하고 있다.

<69> 도 2를 참조하면, 웨이퍼(W) 상에 형성된 미스얼라인먼트 사이트(MS)들 중 측정 사이트를 지정한 후에 지정된 사이트의 하부 패턴층에 형성된 외측 얼라인먼트 마크(OM)와 상부 패턴층에 형성된 내측 얼라인먼트 마크(IM) 사이의 dx/dy 를 측정하게 되고, 이 데이터에 대한 회귀분석을 통해 다음과 같은 미스얼라인먼트 파라미터들을 추출하게 된다.

<70> 1) 웨이퍼 관련 파라미터

<71> 오프셋(Offset); 얼라인먼트 패턴이 좌우, 상하로 어긋난 정도.

<72> 스케일링(Scaling); 렌즈에 의해 웨이퍼 상의 패턴이 좌우, 상하로 확대된 정도.

<73> 회전(W Rotation); 얼라인먼트 패턴의 축이 얼라인먼트 기준축에 대해 틀어진 정도.

- <74> 직교(ORTHogonality); 웨이퍼 얼라인 축이 서로 틀어진 정도.
- <75> 2)레티클에 관련된 파라미터
- <76> 레티클 회전(reticle ROTation);레티클이 부정확하게 세팅되어 얼라인먼트 패턴의 축이 얼라인먼트 기준축에 대해 틀어진 정도.
- <77> 레티클 축소(reticle REDuction);레티클이 부정확하게 세팅되어 웨이퍼 상의 패턴이 좌우 상하로 확대된 정도.
- <78> 도 3에 도시한 바와 같이, 스텝퍼인 노광시스템(40)은 웨이퍼 스테이지(42), 렌즈계(44), 레티클(46) 및 미도시한 광원계를 포함한다. 광원계의 광이 레티클(46)의 마스크 패턴 및 렌즈계(44)를 통하여 웨이퍼(W) 상에 조사되면 웨이퍼 상에 레티클의 마스크 패턴이 축소투영된다.
- <79> 따라서, 웨이퍼 얼라인먼트는 웨이퍼 스테이지(42)에 재치된 웨이퍼(W)의 X축 및 Y축 어긋남, 회전, 직교 등의 보정이 필요하고, 웨이퍼 상에 투영된 상의 좌우 틀어짐 등의 보정이 필요하다. 레티클 얼라인먼트는 레티클의 회전 및 축소 등의 보정이 필요하다.
- <80> 따라서, 오버레이 측정기(20)는 도 4에 도시한 10개의 측정 파라미터, OF-X, OF-Y, SC-X, SC-Y, ORT, W-ROT, RED-X, RED-Y, ROT-X, ROT-Y를 측정시간 및 로트 ID와 함께 오버레이 보정값 제어기(30)에 제공한다.
- <81> 본 발명에 의한 오버레이 보정값 제어기(30)는 노광장치, 즉 스텝퍼의 보정 입력값, $x(n)$ 을 추정하기 위한 Adaptive Least Mean Square Neural Network (ALMS-NN) 알고리즘을 수행한다. 오버레이 보정값 제어기(30)는 장비 입력값을 ALMS-NN 알고리즘에

의해 추정하여 스텝퍼(14)에 도 5에 도시한 데이터를 제공한다. 스텝퍼에 제공되는 데이터는 생성시간 및 룯트 ID 별로 FWD 데이터, RET 데이터, NN 데이터, IN 데이터를 포함한다.

<82> 여기서, FWD 데이터는 KEY 층(LAYER)의 스텝 에러값이고, RET 데이터는 바이어스 데이터이고, NN 데이터는 신경회로망에서 계산된 출력값이고, IN 데이터는 장비 보정값, 즉 장비 입력값이다.

<83> ALMS-NN 알고리즘

<84> 먼저, 포토공정에서 계측 측정오차를 최소화하기 위해서는 장치의 입력값 $x(n)$ 을 추정할 수 있는 알고리즘이 필요하다. 종래의 보정 시스템의 데이터베이스로부터 얻은 $x(n)$ 은 어떠한 해석 가능한 함수로도 근사가 불가능할 정도의 임의성을 보이고 있다.

<85> 도 6은 샘플로 선정한 포토장치의 650 룯트의 데이터들 중 장비 보정값 중 하나인, offset-x에 대한 장비 입력값을 시간순으로 그래프로 나타낸 것이다. 도 7은 도 6의 장비 입력값들을 동일 히스토를 갖는 룯트별로 정렬한 것이며, 도 8은 장비 입력값 중 바이어스값을 빼고 난 후의 값들을 나타낸다.

<86> 도 6 내지 도 8에 도시한 바와 같이, 장비 보정값 $x(n)$ 은 해당 룯트의 히스토리와 밀접한 관계가 있음을 알 수 있으며, 동일 히스토리 별로 일정한 바이어스가 걸려 있음을 알 수 있다.

<87> 동일 히스토리별로 장비 입력값이 차이가 나는 것은 레티클에 찍혀 있는 패턴의 오차, BASE I, BASE II 등의 포토장비의 특성차이 등의 여러 가지 요인들의 작용에 의한

것이다.

<88> 따라서, 장비 입력값은 도 9에 도시한 바와 같이, 해석 가능한 어떠한 함수로도 근사가 불가능할 정도의 임의성을 가짐을 알 수 있다.

<89> 도 10에서와 같이 장비 보정값 $x(n)$ 은 주파수 스펙트럼이 전 주파수 영역에 고르게 퍼져 있는 백색잡음에 가까운 신호특성을 보여 주고 있어서 이를 해석 가능한 시간의 함수로 근사한다는 것이 상당히 어려운 일임을 알 수 있다. 이러한 장비 보정값 $x(n)$ 의 임의적 성질은 오버레이 측정기의 측정오차를 최소화하기 위한 장비 보정값 $x(n)$ 의 추정을 힘들게 하는 주요한 요인이 되고 있다.

<90> 그러나, 장비 보정값 $x(n)$ 의 변화요인에는 정확한 원인을 알 수 없는 임의적 요소들만 있는 것은 아니다. 해당 로트의 히스토리와 연관되는 바이어스 부분은 어느 정도 효과적으로 추정이 가능한 부분이 된다.

<91> ALMS-NN 알고리즘에서는 $x(n)$ 의 효과적인 추정을 위해 $x(n)$ 을 히스토리에 연관성이 있는 바이어스 성분 $x_{bias}(n)$ 과 변화원인을 정확히 알 수 없는 임의적 성분인 랜드성분 $x_{rand}(n)$ 으로 분리해 주고, $x_{bias}(n)$ 은 해당 로트의 히스토리를 바탕으로 추정한다.

<92> $x_{bias}(n)$ 의 정확한 추정을 힘들게 하는 요소인 랜드성분은 그 임의적 성질 때문에 정확한 추정은 거의 불가능하며 다만 그 변화를 신경회로망의 역전파(back propagation) 학습을 이용하여 추종(tracking)하게 함으로써 $x_{bias}(n)$ 의 추정오차를 최소화한다.

<93> $x_{rand}(n)$ 성분은 해당 로트의 히스토리에 대한 연관성이 제거된 상태가 되기때문에 해당 로트의 히스토리에 상관없이 모든 데이터를 이용할 수가 있다.

<94> 따라서, 히스토리의 시간적 변화가 크지 않을 경우 기존 보정 시스템에서 문제가

되었던 데이터 사용기한 제약조건이 해결될 수 있다. 또한 해당 룯트의 히스토리 이외에 $x(n)$ 에 영향을 미치는 외부적 요인이 변화할 경우도 신경회로망이 지닌 학습능력을 이용하여 효과적으로 대처할 수 있게 된다.

<95> 1. $x_{bias}(n)$ 추정 방법

<96> (1) 동일 히스토리가 존재한 경우

<97> $x_{bias}(n)$ 를 추정하는 가장 간단한 방법은 과거의 동일 히스토리를 갖는 룯트들에 대한 장치 입력값들의 평균값을 이용하는 것이다. 어떤 동일 히스토리를 갖는 룯트들의 현재까지 파악된 과거 장치 보정값들을 $x^{sh}(m), m=1,2,\dots,n$ 이라고 하고 이 히스토리에 대한 x_{bias} 값이 상수로 정해져 있다고 가정하자. 그러면, $x^{sh}(m)$ 은 다음과 같이 나타낼 수 있다.

<98>
$$x^{sh}(m) = x_{bias} + x^{sh}_{rand}(m)$$

<99> 이때, $x^{sh}(m)$ 의 완전한 임의성을 가정하면,

<100>
$$\lim_{n \rightarrow \infty} \left(\frac{1}{n} \sum_{m=1}^n x^{sh}_{rand}(m) \right) = 0$$

<101> 이므로

<102>
$$\begin{aligned} \lim_{n \rightarrow \infty} \left(\frac{1}{n} \sum_{m=1}^n x^{sh}(m) \right) &= x_{bias} + \lim_{n \rightarrow \infty} \left(\frac{1}{n} \sum_{m=1}^n x^{sh}_{rand}(m) \right) \\ &= x_{bias} \end{aligned}$$

<103> 결국 $x^{sh}(m)$ 의 평균이 x_{bias} 에 대한 효과적인 평가자로서 역할을 할 수 있음을 알 수 있다.

<104> 그러나, 국소적으로 봤을 때, $x^{sh}_{rand}(m)$ 의 완전한 임의성을 기대할 수 없고 동일 히스토리라 해도 BASE I 장비나 BASE II 장비의 변화, 레티클의 잦은 변경에 따라 발생하는 오차와 같은 요인들을 고려한다면 동일 히스토리 룯트들에 대한 x_{bias} 값도 상수라고 간주하기에는 현실적으로 무리가 있는 것이 사실이다.

<105> 그러므로, 이러한 변화요인을 감안하기 위하여 제안된 ALMS-NN 알고리즘에서는 x_{bias} 추정을 위하여 $x^{sh}(m)$ 을 단순한 산술평균이 아닌 구간 선형 가중평균을 이용한다.

$$x_{bias} = \frac{1}{W} \sum_{i=n-W+1}^n \left[\frac{W+n-i}{\sum_{j=1}^W i} x^{sh}(i) \right]$$

<107> x_{bias} ; 바이어스 보정값

<108> W ; 구간

<109> x^{sh} ; 동일 히스토리를 갖는 이전 바이어스 보정값

<110> (2) 동일 히스토리가 존재하지 않는 경우

<111> 이렇게 동일 히스토리 데이터의 분류와 구간 선형 가중평균을 이용하여 x_{bias} 를 추정하게 되면 히스토리에 대한 제약조건 때문에 동일 히스토리가 존재하지 않는 룯트를

가공해야 하는 경우에 문제가 발생한다.

<112> 물론 한번도 공정을 진행해보지 않은 새로운 디바이스의 경우는 어쩔 수 없이 새로운 데이터의 확보를 위한 샘플링 공정을 피할 수는 없다. 그러나, 디바이스는 같으나 히스토리 구성요소 중 Base I 장치나 Base II 장치나 PPID의 차이 때문에 위와 같은 구간 선형 가중평균을 이용할 수 없는 경우는 다음과 같은 방법들을 통해 x_{bias} 를 추정할 수 있다.

<113> 같은 디바이스라 해도 히스토리에 따라 장비 입력값이 차이가 나는 이유는 당연히 히스토리를 구성하는 각 요소의 특성차이 때문이다. 그러므로, x_{bias} 를 각 히스토리 구성요소(PPID, Base I, Base II, Reticle)에 의한 성분으로 분리해 볼 수 있다. 이들 구성요소들의 상호 연관 관계를 고려하지 않는다면, x_{bias} 는 다음과 같이 나타낼 수 있다.

$$<114> \quad x_{bias} = x_{bias}^{PPID} + x_{bias}^{BASE I} + x_{bias}^{BASE II} + x_{bias}^{RET}$$

<115> x_{bias}^{PPID} , $x_{bias}^{BASE II}$, $x_{bias}^{BASE I}$, x_{bias}^{RET} 은 각각 PPID, BASE I, BASEII, RETICLE에 대응하는 바이어스 성분이다. 가장 큰 영향을 미치는 성분은 포토공정의 기본이 되는 레티클에 의한 성분 x_{bias}^{RET} 이며, 나머지 성분들은 그 영향력이 상대적으로 미미하다고 볼 수 있다. 따라서, 동일 히스토리가 존재하지 않을 경우에 x_{bias}^{RET} 이외의 영향력은 $x_{bias}^{PPID} > x_{bias}^{BASE I} > x_{bias}^{BASE II}$ 순으로 파악된다.

<116> 동일 히스토리가 존재하지 않는 경우에는 다음에 설명할 2가지의 방법을 이용하여 각 성분을 유추한다.

<117> 각 방법에 대한 설명은 적용 우선 순위가 높은 순서로 이루어진다. 그리고, 히스토리는 (PPID, BaseI, BaseII, Reticle) 순으로 나타내기로 한다. 이외의 다른 성분들은

적절한 방법으로 유추해 주어도 큰 무리가 없을 것이다.

<118> (a) Base II, BASE I 또는 PPID 의 상대값을 이용한 x_{bias} 의 유추

<119> 현재 x_{bias}^1 를 추정하고 싶은 로트의 히스토리(H1)와 이전 히스토리 데이터들(H2, H3, H4)이 다음과 같다고 가정하자.

<120>

$$\begin{aligned} H1 &= P1, BI1, BII2, R1, x_{bias}^1 \\ H2 &= P1, BI1, BII3, R1, x_{bias}^2 \\ H3 &= P2, BI4, BII2, R1, x_{bias}^3 \\ H4 &= P2, BI4, BII3, R1, x_{bias}^4 \end{aligned}$$

<121> H1과 H2는 BII2와 BII3만 다르고 나머지 히스토리 구성요소는 동일하고, 마찬가지로 H3과 H4는 BII2와 BII3만 다르고 나머지 히스토리 구성요소는 동일함을 알 수 있다.

<122> 현재 추정하고자 하는 H1에 대한 x_{bias}^1 은 다음과 같이 추정할 수 있다.

<123>

$$x_{bias}^1 - x_{bias}^2 = x_{bias}^3 - x_{bias}^4$$

<124> 따라서, $x_{bias}^1 = x_{bias}^3 - x_{bias}^4 + x_{bias}^2$ 으로 유추할 수 있다.

<125> 영향력이 가장 작은 순서, 즉 BASE II, BASE I, PPID 순으로 유추한다.

<126> x_{bias}^{RET} 성분이 x_{bias} 에 미치는 영향력은 다른 성분들보다 월등히 크기 때문에 H1, H2, H3는 레티클 성분이 H1과 동일한 것만 고려해야 한다.

<127> (b) Base II 또는 BASE I 값만이 다른 히스토리들의 평균값을 이용한 x_{bias} 유추

<128>

현재 x_{bias} 를 추정하고 싶은 롯트의 히스토리가 $H = \{P, BI1, BII2, R\}$ 이라고

하자. 또 H 와 Base II 만 다른 m 개의 히스토리 데이터 $H1 \sim Hm$ 이 존재하고, 각각에 해당하는 x_{bias} 는 x_{bias}^1, x_{bias}^m 이라 하면 다음 수식과 같이 평균값으로 유추한다.

<129>

$$x_{bias} = \frac{1}{m} \sum_{i=1}^m x_{bias}(i)$$

<130>

(3) x_{bias} 를 추정할 수 없는 경우

<131>

만일 현재 공정을 진행하고자 하는 롯트에 대한 동일 히스토리 데이터가 존재하지 않고 상기(2)의 유추방법도 적용할 수 없는 경우에는 샘플링 공정의 진행이 불가피하게 된다.

<132>

2. x_{rand} 추종 방법

<133>

장비 입력값에 영향을 미치는 주된 요소는 각 롯트의 히스토리가 된다.

보정값 $x(n)$ 에서 히스토리를 통해 추정할 수 있는 x_{bias} 값을 제외한 나머지 모든 임의적 요소들은 x_{rand} 에 포함되게 된다.

<134>

이러한 x_{rand} 값은 해당 롯트의 히스토리에 대한 데이터의 의존성이 제거된 상태이므로 여러 가지 외적인 요소로 인한 장비 자체의 시간에 따른 특성변화를 나타낸다고 볼 수 있다.

- <135> 그러나, 현재까지 x_{rand} 의 값을 시간에 따라 변화시키는 요인은 정확하게 밝혀지지 않았기 때문에 x_{rand} 값을 추정한다는 것은 어려운 일이다.
- <136> 이러한 상황에서 적용할 수 있는 효과적인 방법들 중의 하나는 신경회로망을 이용하여 x_{rand} 의 변화를 추종하게 만든 것이다.
- <137> 현재 보정 시스템에서도 이러한 장치 특성의 변화를 포함한 여러 변화요인을 고려하기 위해 같은 히스토리를 갖는 룯트에 대한 데이터들 중에서도 상당히 제한된 기한 내의 데이터만을 이용하고 있다. 그러나 공정 운영에서는 이러한 제한사항들로 인해 문제점들이 야기되고 있어서 ALMS-NN 알고리즘에서는 이러한 변화요인을 히스토리과 분리시켜 독립적으로 고려해 줌으로써 기존 시스템의 단점을 보완하여 공정운영 효율을 높이고자 하였다.
- <138> x_{rand} 추종을 위해 이용한 신경회로망은 도 11에 도시한 바와 같이 3개의 입력 뉴런을 포함한 입력층(50)과 1개의 출력 뉴런을 포함한 출력층(58), 그리고 각각 5개, 5개, 3개의 뉴런으로 이루어진 3층의 은닉층들(52, 54, 56)을 갖는 다층 신경망(Multilayer perceptron)이고, 신경회로망의 학습 방법은 가장 일반적인 오류 역전파(error back propagation) 방식을 이용했다.
- <139> 네트워크는 입력층(50), 은닉층(52~56), 출력층(58) 방향으로 연결되어 있으며, 각 층 내의 연결과 출력층(58)에서 입력층(50)으로의 직접적인 연결은 존재하지 않는 전방향(feedforward) 네트워크이다.
- <140> ALMS-NN 알고리즘에서 이용한 오류역전파 학습방법에서는 입력층(50)의 각 노드에 입력패턴을 주면, 이 신호는 각 노드에서 변환되어 은닉층(52, 54, 56)에 전달되고, 최

후에 출력층(58)으로 출력된다. 이 출력값과 기대값을 비교하여 그 차이를 줄여나가는 방향으로 연결강도(웨이트)(w)를 조절하고, 상위층에서 역전파하여 하위층에서는 이를 근거로 다시 자기층의 연결강도(웨이트)(w)를 조정해 나간다.

<141> 신경회로망의 전향경로를 통한 시스템의 출력은 다음과 같다.

$$<142> \quad a^{m+1} = f^{m+1}(w^{m+1}a^m + b^{m+1})$$

<143> 여기서, m은 0, 1, ... M-1 이고, a는 각 뉴런의 출력, b는 바이어스, f는 전달함수, w는 웨이트를 나타낸다. M은 신경망의 층수를 나타낸다.

<144> 다층신경망의 오류 역전파 알고리즘에서 기대값과 차에 의한 오차는 다음과 같다.

$$<145> \quad F(X) = (t(k) - a(k))^T(t(k) - a(k)) = e^T(k)e(k)$$

<146> 여기서 F는 자승오차를 나타내고, X는 신경망의 웨이트와 바이어스 벡터를 나타내고, t는 기대 출력값을 나타낸다.

<147> 여기서 근사화된 최급강하 알고리즘을 행렬식으로 표현하면 다음과 같다.

$$<148> \quad w^m(k+1) = w^m(k) - as^m(a^{m-1})^T b^m(k+1) = b^m(k) - as^m$$

<149> 여기서 F의 변화를 감도(sensitivity)로 하고 행렬식으로 표현하면 다음과 같다.

$$<150> \quad s^m = F^m(n^m)(w^{m+1})^T s^{m+1}$$

<151> 감도는 신경망의 마지막 층에서 시작하여 첫 번째 단으로 다음과 같이 역으로 전파된다.

<152>

$$s^m \rightarrow s^{M-1} \rightarrow \dots \rightarrow s^2 \rightarrow s^1$$

<153>

여기서 첫단의 감도는 아래 수식으로 표현된다.

<154>

$$s^m = -2F^M(n^M)(t-a)$$

<155>

신경회로망을 이용하여 x_{rand} 의 변화를 추종하도록 하기 위하여 다음 수식과 같은 입출력관계를 고려한다.

<156>

$$x_{rand}(n) = f(x_{rand}(n-1), x_{rand}(n-2), x_{rand}(n-3))$$

<157>

즉 신경회로망으로 하여금 가장 최근의 x_{rand} 값 3개 데이터를 참고해서 현재의 x_{rand} 값의 추정값을 생성한다. 생성된 추정값은 나중에 오버레이 측정기를 통해 측정된 실제값과 오차를 줄이는 방향으로 신경회로망을 학습시키는 과정을 통해 신경회로망의 출력이 x_{rand} 값의 변화를 추종하도록 한다.

<158>

은닉층(52, 54, 56) 뉴런들의 전달함수로는 다음 수식으로 표현되는 시그모이드 (sigmoid) 함수를 이용한다.

<159>

$$\phi(v) = \frac{1}{1 + \exp(-\alpha v)}$$

<160>

출력 뉴런의 전달함수는 선형함수를 사용한다.

<161>

신경회로망의 일반화능력과 학습능력을 증가시키기 위해 은닉층을 갖도록 하였으나 은닉층의 수와 신경회로망의 성능이 반드시 비례관계에 있는 것만은 아니며, 3층 이상의 은닉층은 신경회로망의 성능향상에 큰 도움이 되지 못한다는 것은 일반적으로 알려진 사

실이다.

- <162> 3. Photo 장치 입력값 생성방법 및 신경회로망의 학습방법
- <163> 도 12를 참조하면, 본 발명의 시스템은 먼저 시스템의 ALMS-NN 변수 및 신경망 셋업 변수 등을 초기화한다(S102). 이어서 실행모드를 입력받고(S104) 입력된 실행모드가 셋업모드인지 로드모드인지를 판단한다(S106).
- <164> S106단계에서 셋업모드이면 셋업파일을 오픈하여 하나의 레코드를 읽어서 상기 초기화된 각 변수에 대입하고(S108), 셋업모듈을 수행한다(S110).
- <165> S106단계에서 셋업모드가 아니면 로드모드인지를 체크한다(S112). S112단계에서 로드모드이면 로드파일명을 입력하고 파일을 로딩한다(S114).
- <166> S110단계 및 S114단계를 수행한 다음에는 로드모듈을 수행한다(S116).
- <167> 도 13a 및 도 13b를 참조하면, 셋업모듈은 셋업파일이 종료인지를 체크하고, 종료이면 결과를 출력하고(S120) 메인 프로그램으로 리턴한다.
- <168> S118단계에서 파일 종료가 아니면 임의 변수에 저장된 데이터를 셋업모듈에 사용될 변수에 대입한다(S122). 대입된 변수 중 히스토리 구성요소인 변수들의 값들을 사용하여 동일 히스토리 데이터가 존재하는지를 검색한다(S124). 그리고, 대입된 변수들로부터 장비 순수 보정값을 계산하고(S126) 신경망 학습모듈을 수행한다(S128).
- <169> S128단계에서 신경망 학습 후에 현재 학습된 데이터의 동일 히스토리가 존재하지 않을 경우에는(S130) 새로운 바이어스 데이터로 추가하고(S132) S118단계를 수행한다.
- <170> S130단계에서 동일 히스토리 데이터가 존재하는 경우에는 동일 히스토리 바이어스

데이터가 윈도우 사이즈인가를 판단한다(S134). S134단계에서 윈도우 사이즈인 경우에는 윈도우 사이즈로 바이어스 평균값을 구하고(S136) S118단계를 수행한다. S134단계에서 윈도우 사이즈가 아닌 경우에는 바이어스의 평균값을 구하고(S138) S118단계를 수행한다.

<171> 도 14를 참조하면, 로드모듈에서는 실행모드를 선택하고(S140), 선택된 3가지 실행 모드에 따라 REQUEST 모듈(S142), FEEDBACK 모듈(S144), MODIFY 모듈(S146)을 각각 수행한다.

<172> 도 15를 참조하면, 리퀘스트 모듈(S142)에서는 새로운 롯데가 장비에 트랙인(track-in)되면 롯데 데이터를 읽어서 리퀘스트 모듈에서 사용될 변수에 대입한다(S148). ALMS-NN 알고리즘에서의 우선 롯데의 히스토리 구성요소, 즉 PPID, Base I, Base II, Reticle을 확인하고 과거에 동일한 히스토리를 갖는 롯데들을 가공했을 때의 데이터를 찾는다(S150).

<173> 만일 동일 히스토리 데이터들이 존재할 경우에는 최근 10개의 바이어스 데이터를 읽어와서(S152) 이들 데이터에 대한 구간 선형 가중평균으로 해당 롯데에 대한 x_{bias} 의 추정값인 \hat{x}_{bias} 를 구한다.

<174> S150단계에서 동일 히스토리 데이터가 존재하지 않을 경우에는 상술한 2단계의 유추방법에 의해 \hat{x}_{bias} 을 구하기 위하여 바이어스 유추모듈을 수행한다(S154).

<175> S152 및 S154에서 바이어스값을 추정하고 랜드값을 추정하기 위하여 신경망 모듈을 수행하여(S156) 장비 입력값을 추정한다.

<176> 최근 3개의 랜드 보정값을 기반으로 신경회로망이 추정한 \hat{x}_{rand} 을 구한다.

<177> 최종 장비 입력값은 다음 식에 의해 결정된다.

<178>

$$\hat{i}(n) = \hat{x}_{bias} + \hat{x}_{rana} + f(n)$$

<179>

여기서 $f(n)$ 은 피드포워드(feedforward) 입력값이다.

<180>

직교(orthogonality)의 경우는 다른 입력 파라미터들과는 달리 보정값이

<181>

$x(n) = i(n) + f(n) + e(n)$ 와 같이 주어지기 때문에 x_{bias} 를 추정할 때나 입력값을 생성할 때 약간의 차이가 생기게 된다.

<182>

따라서, 직교 성분의 장비 입력값은 다음 수식으로 표현된다.

<183>

$$\hat{i}(n) = \hat{x}_{bias} + \hat{x}_{rana} - f(n)$$

<184>

신경망을 학습하기 위해서는 피드백된 오버레이 측정기의 측정값을 이용한다. 어떤 룯트에 대한 측정기의 측정오차를 0으로 만들 수 있는 이상적인 장비입력을 $i(n)$ 이라 할 때, 결국 $\hat{i}(n)$ 과 $i(n)$ 의 차이가 측정오차로 나타나게 되므로 수식적으로 살펴 보면 다음과 같다.

<185>

$$x(n) = \hat{i}(n) - f(n) - e(n) = i(n) - f(n)$$

<186>

결국 $\hat{i}(n) - i(n) = e(n)$ 이 되어 이를 신경회로망의 오류 역전파 학습에 이용한다.

<187>

직교성분의 경우에는 $e(n)$ 을 그대로 피드백시키는 것이 아니라 $-e(n)$ 을 피드백시켜야 한다. 따라서,

<188>

$$x(n) = \hat{i}(n) + f(n) + e(n) = i(n) + f(n) \text{이므로 결국 } \hat{i}(n) - i(n) = -e(n) \text{이 되는 것이다.}$$

- <189> S156단계 이전에서 바이어스값을 추정할 수 없는 경우에는 샘플링 여부를 판단하고(S158) 샘플링 공정이 필요없는 경우에는 현재 룯트에 대한 입력값을 파일로 출력하고 데이터를 저장한다(S160).
- <190> S158단계에서 샘플링 공정이 필요하다고 판단되면, 샘플링 진행을 권고하는 메시지를 출력하고 샘플링 진행 데이터를 저장한다(S162).
- <191> 도 16을 참조하면, 피드백 모듈에서는 피드백파일에서 데이터를 읽어온다(S164). 피드백 데이터의 값이 SPEC-IN RANGE 보다 5배 이상인가 아닌가를 판단하고(S166), 5배 이상이면 현재 피드백된 데이터와 동일 데이터를 임시 파일 데이터(ASS-DATA)에서 삭제한다(S168).
- <192> S166단계에서 5배 미만인 경우에는 ASS-DATA에서 현재 피드백된 데이터와 동일 룯트 데이터를 찾고(S170), 해당 룯트에 대한 히스토리 존재여부를 판단하여 신경 회로망 업데이트를 위한 순수 보정값을 계산한다(S172).
- <193> 이어서, 신경망 데이터를 업데이트하고(S174), 신경망 업데이트 모듈을 수행한 후에 메인 프로그램으로 리턴한다(S176).
- <194> 도 17을 참조하면, 모디파이 모듈에서는 ASS-DATA를 출력하여 모니터 상에 디스플레이한다(S178). 이에 오퍼레이터에 의한 지울 룯트 번호를 입력받고(S180) 입력된 해당 룯트 변수를 ASS-DATA에서 삭제하고(S182), 삭제된 결과를 모니터 상에 디스플레이한다(S184).
- <195> 도 18a 및 도 18b를 참조하면, 바이어스 유추모듈에서는 먼저 히스토리 구성요소 중 영향력이 가장 작은 BASE II 성분만 다르고 나머지 3개의 히스토리 구성요소가 동일

한 바이어스 데이터를 추출한다(S186). S186단계에서 추출된 데이터가 존재하는 지를 체크하고(S188) 존재할 경우에는 현재 룩트와 동일한 바이어스 데이터를 추출한다(S190). 추출된 바이어스 데이터들을 조합하여 상술한 바와 같이 BASE II의 상대 바이어스를 구한다(S192).

<196> S192단계에서 상대 바이어스가 구해질 경우에는 종료하고 리퀘스트 모듈로 리턴한다(S194). S192단계에서 상술한 공식에 의한 상대 바이어스가 구해지지 않거나 S188단계에서 바이어스 데이터의 추출이 없을 경우에는 BASE I만 다르고 나머지 3개의 히스토리가 동일한 바이어스 데이터를 추출한다(S196).

<197> S196단계에서 추출된 데이터가 존재하는지를 체크하고(S198) 존재할 경우에는 현재 룩트와 동일한 바이어스 데이터를 추출한다(S200). 추출된 바이어스 데이터들을 조합하여 상술한 바와 같이 BASE I의 상대 바이어스를 구한다(S202).

<198> S202단계에서 상대 바이어스가 구해질 경우에는 종료하고 리퀘스트 모듈로 리턴한다(S204). S202단계에서 상대 바이어스가 구해지지 않거나 S198단계에서 바이어스 데이터의 추출이 없을 경우에는 PPID만 다르고 나머지 3개의 히스토리가 동일한 바이어스 데이터를 추출한다(S206).

<199> S206단계에서 추출된 데이터가 존재하는지를 체크하고(S208) 존재할 경우에는 현재 룩트와 동일한 바이어스 데이터를 추출한다(S210). 추출된 바이어스 데이터들을 조합하여 상술한 바와 같이 PPID의 상대 바이어스를 구한다(S214).

<200> S214단계에서 상대 바이어스가 구해질 경우에는 종료하고 리퀘스트 모듈로 리턴한다(S216). S214단계에서 상술한 공식에 의한 상대 바이어스가 구해지지 않을 경우에는

각각 BASE II, BASE I 중 어느 하나만 다르고 나머지 3개의 히스토리가 동일한 바이어스 데이터들의 평균값을 구한다(S212).

<201> 도 19를 참조하면, 신경망 모듈에서는 신경망의 입력층 데이터를 업데이트하고 (S214), 1차 은닉층 출력(S216), 2차 은닉층 출력(S218), 3차 은닉층 출력(S220)을 거쳐서 출력층을 통해 최종 출력을 산출한다(S222). 산출된 랜드 보정값과 바이어스값을 합산하여 장비 보정값을 계산한다(S224).

<202> 도 20을 참조하면, 신경망 업데이트 모듈에서는 오류 역전파 학습을 위하여 3차 은닉층부터 1차은닉층까지 차례로 각 층의 감도를 계산한다(S226~S230). 이어서, 계산된 감도에 응답하여 3차 은닉층부터 1차 은닉층까지 차례로 각 층의 웨이트를 업데이트하여 (S232~S236) 신경망의 업데이트를 종료한다.

<203> * 모의 실험 결과

<204> 모의실험은 A, B, C의 3대의 포토장비 데이터에 대해 실시한다. 장치를 처음 시동했을 경우를 가정하여 과거 진행데이터가 전무한 상태에서 모의실험을 시작했으며, 기존 알고리즘과의 성능비교를 통해 ALMS-NN 알고리즘의 성능을 검증하였다.

<205> 또, 실제 공정 운영시는 포토 공정이 끝난 후 바로 측정이 이루어지지 않는기 때문에 모의실험은 이를 고려하여 5 롯트 정도의 측정지연을 가정하여 실시했다. 즉 공정을 진행하고 난 후 5 Lot 정도의 공정시간이 지난 후에 측정오차를 이용할 수 있도록 하였다.

<206> 성능비교를 위해 살펴본 모의실험 결과는 측정오차가 spec-in 범위를 벗어나지 않는 비율인 spec-in ratio이다. 기존 알고리즘의 spec-in ratio는 데이터상의 측정오차를

이용하여 구했고, ALMS-NN 알고리즘에 대해서 ALMS-NN 알고리즘에서 생성된 장치 입력값을 적용했을 경우 측정될 것이라고 생각되는 가상의 측정오차를 구하여 spec-in ratio를 구했다.

<207> 기존 보정 시스템에서 사용했던 장치 입력값을 $i_e(n)$, 이를 이용하여 공정을 진행한 롯트에서 측정된 오버레이 측정오차를 $e_e(n)$ 이라 하고, ALMS-NN 알고리즘에서 계산된 장비 입력값을 $i_a(n)$, 이를 이용했을 경우에 측정될 가상 측정오차를 $e_a(n)$ 이라고 하면, 다음 수식이 성립된다.

<208>

$$i_e(n) - e_e(n) = i_a(n) - e_a(n)$$

<209> 그리고, 기존 알고리즘을 이용했을 경우 샘플공정을 진행할 수 없는 경우들에 대해 ALMS-NN 알고리즘을 적용했을 경우의 spec-in ratio를 살펴봄으로써 샘플공정 회수 감소 가능성을 검증해 보았다.

<210> 기존 시스템에서 샘플공정을 진행해야 되는 경우들로는, 현재 진행하고자 하는 Lot과 동일한 히스토리를 갖는 Lot을 가공한 적이 없어 참고할 데이터가 없는 경우(no history case)와 동일 히스토리를 갖는 Lot을 가공한 적은 있으나 정해진 기한을 지나 그 데이터를 신뢰할 수 없는 경우(old history case)이다.

<211> 시뮬레이션에서 고려한 기한은 150 Lot 가공시간이다. 즉, 동일 히스토리 데이터가 존재하나 최근 150 Lot에는 존재하지 않는 경우를 old history case로 보았다. 하나의 Lot을 가공하는데 드는 시간은 대략 40분이지만 모든 Lot에 대해 측정이 이루어지는 것은 아니라는 점을 고려했을 때, 모의실험에서 사용한 데이터는 모두 측정이 이루어진 경우만을 이용하므로 1 Lot 가공시간을 대략 1시간으로 생각해도 큰 무리가 없을 것으로

보인다.

<212> 따라서, 150 롯트 정도의 가공시간이면 대략 5 내지 6일 정도의 시간으로 볼 수 있다. 기존 알고리즘에서 사용할 수 있는 데이터의 기한은 3~5일이므로 최근 진행한 150 Lot 정도에 현재 Lot과 동일한 히스토리를 갖는 Lot이 존재하지 않으면 기존 알고리즘에서는 샘플공정을 진행해야 하는 경우들이다.

<213> Spec-in ratio를 측정하기 위해 Offset-X, Offset-Y의 경우 $-0.03 \sim 0.03$, Scale-X, Scale-Y, Orthogonality, Wafer rotation-Y의 경우는 $-0.3 \sim 0.3$, 그리고 Reticle reduction, Reticle rotation의 경우는 $-1.5 \sim 1.5$ 의 spec-in 범위를 적용하였다.

<214> < A 호기 장치에 대한 모의실험 결과 >

<215> 데이터 수 총 5200개 중 No history cases는 40회가 발생되었고, 그 중에서 x_{bias} 를 유추하여 사용한 경우는 13회이고, Old history cases는 99회가 발생되었다.

<216> * 전체 Spec-in ratio(%)

<217>

	ALMS-NN	기존 시스템
offset-x	91.01	89.27
offset-y	89.64	88.35
scale x	87.49	86.46
scale y	83.24	84.15
orthogonality	81.87	79.15
wafer rotation	89.76	87.69
reticle-reduction	94.97	94.15
reticle-rotation	86.31	82.08

<218> * x_{bias} 를 유추해 사용한 경우의 spec-in ratio(%)

<219>

x_{bias} 유추	
offset-x	92.30
offset-y	30.77
scale x	84.62
scale y	69.23
orthogonality	76.92
wafer rotation	92.30
reticle-reduction	76.92
reticle-rotation	69.23

<220> * old history case의 spec-in ratio(%)

<221>

old history	
offset-x	82.83
offset-y	77.78
scale x	92.93
scale y	84.85
orthogonality	84.85
wafer rotation	87.88
reticle-reduction	89.90
reticle-rotation	85.86

<222> < B 호기 장치에 대한 모의실험 결과 >

<223> 데이터 수 총 3400개 중 No history cases는 63회가 발생되었고, 그 중에서 x_{bias} 를 유추하여 사용한 경우는 13회이고, Old history cases는 161회가 발생되었다.

<224> * 전체 Spec-in ratio(%)

<225>

	ALMS-NN	기존 시스템
offset-x	92.21	87.85
offset-y	96.54	93.06
scale x	98.18	97.20
scale y	96.57	94.68
orthogonality	92.87	90.06
wafer rotation	98.00	96.53
reticle-reduction	99.76	99.62
reticle-rotation	99.37	98.21

<226> * x_{bias} 를 유추해 사용한 경우의 spec-in ratio(%)

<227>

x_{bias} 유추	
offset-x	69.23
offset-y	30.77
scale x	100.00
scale y	76.92
orthogonality	92.55
wafer rotation	96.89
reticle-reduction	98.76
reticle-rotation	96.27

<228> * old history case의 spec-in ratio(%)

<229>

old history	
offset-x	91.30
offset-y	95.03
scale x	95.65
scale y	91.92
orthogonality	92.55
wafer rotation	96.89
reticle-reduction	98.76
reticle-rotation	96.27

<230> < C 호기 장치에 대한 모의실험 결과 >

<231> 데이터 수 총 3000개 중 No history cases는 71회가 발생되었고, 그 중에서 x_{bias} 를 유추하여 사용한 경우는 23회이고, Old history cases는 70회가 발생되었다.

<232> * 전체 Spec-in ratio(%)

<233>

	ALMS-NN	기존 시스템
offset-x	95.86	95.43
offset-y	94.68	94.67
scale x	93.84	94.07
scale y	82.36	87.77
orthogonality	86.97	88.63
wafer rotation	89.33	90.23
reticle-reduction	92.39	96.07
reticle-rotation	96.23	94.13

<234> * x_{bias} 를 유추해 사용한 경우의 spec-in ratio(%)

<235>

x_{bias} 유추	
offset-x	82.61
offset-y	56.52
scale x	91.30
scale y	86.96
orthogonality	78.26
wafer rotation	69.57
reticle-reduction	82.61
reticle-rotation	82.61

<236> * old history case의 spec-in ratio(%)

<237>

old history	
offset-x	88.57
offset-y	85.71
scale x	87.14
scale y	87.14
orthogonality	90.00
wafer rotation	76.29
reticle-reduction	75.71
reticle-rotation	88.57

<238> 상술한 모의실험 결과를 종합하면, spec-in ratio의 경우 ALMS-NN 알고리즘이 기존 알고리즘에 비해 다소 향상된 결과를 보이고 있음을 알 수 있다.

- <239> 그러나, ALMS-NN 알고리즘을 제안한 가장 주된 목적은 spec-in ratio의 향상보다는 다품종 소량 생산체제에서 큰 문제가 되고 있는 샘플공정의 회수를 줄이는데 있었음을 고려할 때, 모의실험 결과에서 보이고 있는 ALMS-NN 알고리즘의 spec-in ratio는 어느 정도 만족스러운 결과라고 볼 수 있다.
- <240> 오버레이 측정기의 측정오차의 산포도를 보기 위해 A 호기의 경우 측정기 측정오차의 분포를 도 21 내지 도 25에서 그래프에 나타내 보았으나 기존 알고리즘과 ALMS-NN 알고리즘간의 큰 차이는 없어 보인다. (B, C호기에 대한 계측 측정오차 분포는 A호기와 유사하므로 생략한다.)
- <241> 실제 샘플공정 회수를 줄이기 위해서는 동일 히스토리 데이터가 존재하지 않는 경우에 ALMS-NN 알고리즘을 적용해서 x_{bais} 값을 유추하여 이용한 경우나 오래된 히스토리 데이터를 이용하였을 경우의 spec-in ratio가 전체 spec-in ratio와 비교했을 때 많이 뒤쳐져서는 안된다.
- <242> 동일 히스토리가 존재하지 않는 모든 경우들에 대해 x_{bais} 값을 유추할 수 있는 경우의 확률은 A 호기가 32.5%, B호기가 20.1%, C호기가 32.4% 정도였으며, 이렇게 유추한 값을 이용하여 생성된 장비 입력값으로 공정을 진행했다고 가정했을 때의 spec-in ratio는 전체 spec-in ratio 보다는 다소 떨어지지만 대략 70~80% 정도로 확보됨을 알 수 있다.
- <243> 오래된 히스토리 데이터를 이용하였을 경우에 대한 spec-in ratio는 모의실험을 실시한 세 장치 모두 전체 spec-in ratio와 거의 대등한 결과를 보이고 있다. 이는 ALMS-NN 알고리즘을 적용한다면 기존 알고리즘에서 큰 문제가 되었던 데이터 사용기한

문제를 해결할 수 있는 가능성을 보여 주는 결과라 할 수 있다.

<244> * 실제 장치에 적용 결과

<245> ALMS-NN 알고리즘의 실제 공정에서의 적합성 및 타당성 검증을 위한 공정 적용은 3일에 걸쳐서 실시하였고 테스트를 통해 총 67롯트의 공정을 진행했다.

<246> 테스트 공정을 진행하는 동안 동일 히스토리가 존재하지 않아 x_{basis} 를 유추하는 알고리즘을 적용한 경우는 없었으며, 기존 알고리즘을 이용했다면 데이터의 기한 때문에 샘플공정을 진행해야 했던 경우가 17회 발생했다. 물론, ALMS-NN 알고리즘의 경우 데이터의 기한을 고려하지 않고 적용했기 때문에 이러한 경우도 샘플공정 없이 그대로 공정을 진행했다.

<247> 테스트 결과 spec-in ratio는 약 98%정도를 얻을 수 있었다. 즉, spec-out이 발생하여 재공정을 해야 했던 경우가 약 2%정도 발생한 것이며, 이는 ALMS-NN 알고리즘을 적용하지 않고 기존 알고리즘을 적용했을 때 전체 spec-out 발생율인 8%보다는 월등히 낮은 수치였다.

<248> 도 26 내지 도 35는 테스트를 통해 얻은 ALMS-NN 알고리즘을 적용한 측정기 측정오차(A-KLA)와 기존 알고리즘에서 생성한 장치 입력값을 적용했다고 가정했을 때의 가상의 측정기 측정오차(E-KLA)의 비교 그래프와 ALMS-NN 알고리즘에서 생성된 장치 입력값의 구성을 나타내는 그래프들이다.

<249> 도 26 내지 도 30에 도시한 바와 같이, 오버레이 측정기 측정오차의 산포는 ALMS-NN 알고리즘을 이용하였을 경우나 기존 알고리즘을 이용한 경우가 거의

. 비슷하였다.

<250> 도 31 내지 도 35에 도시한 바와 같이, 장비 입력값은 바이어스 값과 랜드값(신경 회로망 출력값)의 합으로 일치됨을 알 수 있다.

<251> 상술한 실시예에서는 노광장치에 대한 예를 들어 설명하였으나 반도체 공정에서 컴퓨터를 이용한 자동제어방식의 모든 장비, 예컨대 플라즈마 장비, CMP 장비, CVD 장비 등에 대해 과거의 히스토리에 근거한 바이어스 보정값과 임의적인 랜드값으로 구분하여 장비 보정값을 제어함에 있어서 본 발명의 알고리즘의 적용이 가능하다고 할 것이다.

<252> 상술한 바와 같이 본 발명은 기재된 실시예에 대해서만 상세히 설명되었지만 본 발명의 기술사상 범위 내에서 다양한 변형 및 수정이 가능함은 당업자에게 있어서 명백한 것이며, 이러한 변형 및 수정이 첨부된 특허청구범위에 속함은 당연한 것이다.

【발명의 효과】

<253> 상술한 바와 같이 본 발명에서는 포토공정에서 오버레이 측정오차의 정확한 예측과 오차를 보정하기 위한 장비 입력값의 결정을 바이어스 성분과 랜드 성분으로 분리하고 바이어스 성분은 히스토리 추정 및 유추에 의해 결정하고 랜드성분은 신경회로망을 이용한 학습과 추종에 의해 시간제한 없이 모든 데이터를 활용토록 함으로써 샘플링 공정 수를 대폭적으로 감소시킬 수 있어서 생산수율을 향상시킬 수 있다.

【특허청구범위】**【청구항 1】**

피처리물을 처리하기 위한 공정장치의 입력값과, 상기 공정장치에서 처리된 피처리물을 측정하여 얻은 측정값 사이의 오차값을 구하고, 상기 오차값을 줄이는 방향으로 상기 공정장치의 입력값을 보정하기 위한 보정값을 산출하고, 상기 값들을 다음 보정값 산출시에 사용하기 위한 공정 데이터로 관리하는 공정장치의 제어방법에 있어서,

상기 공정장치에 로딩되는 피처리물과 동일 히스토리를 갖는 이전 공정 데이터를 검색하는 단계;

상기 검색된 동일 히스토리를 갖는 이전 공정 데이터들 중 최근 복수개의 이전 보정값들로부터 현재 바이어스 보정값을 추정하는 단계;

이전 공정 데이터들 중 가장 최근 복수개의 이전 랜드 보정값들을 기반으로 신경회로망에 의해 현재 랜드 보정값을 추정하는 단계;

상기 공정장치의 현재 보정값으로 상기 추정된 바이어스 보정값과 랜드 보정값을 합산하는 단계; 및

상기 오차값을 사용하여 상기 랜드 보정값의 변화를 추종하도록 상기 신경회로망을 학습시키는 단계를 구비하는 것을 특징으로 하는 공정장치의 제어방법.

【청구항 2】

제 1 항에 있어서, 상기 현재 바이어스 보정값 추정단계는 다음 수식으로 정의된 구간선형 가중평균 알고리즘에 의해 추정되는 것을 특징으로 하는 공정장치의 제어방법.

$$x_{bias} = \frac{1}{W} \sum_{i=n-W+1}^n \left[\frac{W+n-i}{\sum_{j=1}^W i} x_{sh}(i) \right]$$

x_{bias} ; 바이어스 보정값

W ; 구간

x_{sh} ; 동일 히스토리를 갖는 이전 바이어스 보정값

【청구항 3】

제 1 항에 있어서, 상기 신경회로망은 다층 신경망으로 구성되고, 학습법은 오류 역전파 방식인 것을 특징으로 하는 공정장치의 제어방법.

【청구항 4】

제 1 항에 있어서, 상기 공정장치는 다품종 소량생산 체제의 반도체 장치의 제조장치인 것을 특징으로 하는 공정장치의 제어방법.

【청구항 5】

웨이퍼 상의 포토레지스트를 노광하기 위한 노광장치의 입력값과, 상기 노광장치에서 노광처리되어 현상된 포토레지스트 패턴을 오버레이 측정장치를 통하여 측정하여 얻은 측정값과의 오차값을 구하고, 상기 오차값을 줄이는 방향으로 상기 입력값을 보정하기 위한 보정값을 산출하고, 상기 값들을 다음 보정값 산출시에 사용하기 위한 노광공정 데이터를 생성시간단위로 관리하는 노광장치의 제어방법에 있어서,

상기 노광장치에 로딩되는 새로운 룯트와 동일 히스토리를 갖는 이전 공정 데이터를 검색하는 단계;

상기 검색된 동일 히스토리를 갖는 이전 공정 데이터들 중 최근 복수개의 이전 보정값들로부터 현재 보정값의 바이어스 성분을 추정하는 단계;

이전 공정 데이터들 중 가장 최근 복수개의 이전 랜드 보정값들을 기반으로 신경 회로망에 의해 현재 보정값의 랜드 성분을 추정하는 단계;

상기 노광장치의 현재 보정값으로 상기 추정된 바이어스 성분과 랜드 성분을 합산하는 단계; 및

상기 오차값을 사용하여 상기 랜드 성분의 변화를 추종하도록 상기 신경회로망을 학습시키는 단계를 구비하는 것을 특징으로 하는 노광장치의 제어방법.

【청구항 6】

제 5 항에 있어서, 상기 현재 보정값의 바이어스 부분 추정단계는 다음 수식으로 정의된 구간선형 가중평균 알고리즘에 의해 추정되는 것을 특징으로 하는 노광장치의 제어방법.

$$x_{bias} = \frac{1}{W} \sum_{i=n-W+1}^n \left[\frac{W+n-i}{\sum_{j=1}^W i} x_{sh}(i) \right]$$

x_{bias} ; 보정값의 바이어스 성분

W ; 구간

x_{sh} ; 동일 히스토리를 갖는 이전 바이어스 성분

【청구항 7】

제 6 항에 있어서, 상기 구간은 10으로 하는 것을 특징으로 하는 노광장치의 제어 방법.

【청구항 8】

제 5 항에 있어서, 상기 검색단계에서 히스토리 구성요소인 레티클, PPID, 베이스 I, 베이스 II가 모두 동일한 데이터를 동일 히스토리 공정 데이터로 검출하는 것을 특징으로 하는 노광장치의 제어방법.

【청구항 9】

제 8 항에 있어서, 상기 검색단계에서 동일 히스토리의 공정 데이터가 존재하지 않을 경우에는 상기 레티클 요소가 동일한 공정 데이터들 중에서 나머지 요소들의 우선순위에 따라 보정값의 바이어스 부분을 유추하는 것을 특징으로 하는 노광장치의 제어방법.

【청구항 10】

제 9 항에 있어서, 상기 유추방법은

히스토리 구성요소 중 어느 하나의 구성요소만 다른 공정 데이터들을 추출하는 단계;

상기 추출된 하나의 구성요소만 다른 공정 데이터들 중 어느 하나의 구성요소의 상대값을 이용하여 보정값의 바이어스 성분을 유추하는 단계;

상기 상대값을 이용하여 바이어스 성분을 산출할 수 없을 경우에는 상기 추출된 어느 하나의 구성요소만 다른 공정 데이터들의 평균값을 구하여 보정값의 바이어스 성분으로 유추하는 단계를 구비한 것을 특징으로 하는 노광장치의 제어방법.

【청구항 11】

제 10 항에 있어서, 상기 우선순위는 각 성분의 보정값에 미치는 영향력이 적은 순서인 BASEII, BASEI, PPID 순인 것을 특징으로 하는 노광장치의 제어방법.

【청구항 12】

제 10 항에 있어서, 상기 바이어스 성분을 유추할 수도 없는 경우에는 샘플링 공정을 진행을 요청하는 단계를 더 구비하는 것을 특징으로 하는 노광장치의 제어방법.

【청구항 13】

제 5 항에 있어서, 상기 신경회로망은 다층 신경망으로 구성되고, 학습법은 오류 역전파 방식인 것을 특징으로 하는 노광장치의 제어방법.

【청구항 14】

제 13 항에 있어서, 상기 다층 신경망은

3 개의 입력노드를 가진 입력층;

1개의 출력노드를 가진 출력층; 및

상기 입력층과 출력층 사이에 3층의 은닉층을 구비한 것을 특징으로 하는 노광장치의 제어방법.

【청구항 15】

제 14 항에 있어서, 상기 은닉층 뉴런들은 전달함수로 시그모이드 함수를 사용하는 것을 특징으로 하는 노광장치의 제어방법.

【청구항 16】

제 14 항에 있어서, 상기 출력층 뉴런은 전달함수로 선형함수를 사용하는 것을 특징으로 하는 노광장치의 제어방법.

【청구항 17】

웨이퍼 상의 포토레지스트를 노광하기 위한 노광장치의 입력값과, 상기 노광장치에서 노광처리되어 현상된 포토레지스트 패턴을 오버레이 측정장치를 통하여 측정하여 얻은 측정값과의 오차값을 구하고, 상기 오차값을 줄이는 방향으로 상기 입력값을 보정하기 위한 보정값을 산출하고, 상기 값들을 다음 보정값 산출시에 사용하기 위한 노광공정 데이터를 생성시간단위로 관리하는 노광장치의 제어방법에 있어서,

상기 노광장치에 로딩되는 새로운 룯트와 동일 히스토리 구성요소(레티클, PPID, BASEI, BASEII)를 갖는 이전 공정 데이터를 검색하는 단계;

상기 검색된 동일 히스토리를 갖는 이전 공정 데이터들 중 최근 복수개의 이전 보정값들로부터 현재 보정값의 바이어스 성분을 추정하는 단계;

상기 동일 히스토리를 갖는 이전공정 데이터가 존재하지 않을 경우에는 상기 히스토리 구성요소 중 레티클을 제외한 어느 하나의 구성요소만 다른 공정 데이터들을 추출하는 단계;

상기 추출된 하나의 구성요소만 다른 공정 데이터들 중 어느 하나의 구성요소의 상대값을 이용하여 보정값의 바이어스 성분을 유추하는 단계;

상기 상대값을 이용하여 바이어스 성분을 산출할 수 없을 경우에는 상기 추출된 어느 하나의 구성요소만 다른 공정 데이터들의 평균값을 구하여 보정값의 바이어스 성분으로 유추하는 단계;

이전 공정 데이터들 중 가장 최근 복수개의 이전 랜드 보정값들을 기반으로 신경회로망에 의해 현재 보정값의 랜드 성분을 추정하는 단계;

상기 노광장치의 현재 보정값으로 상기 추정된 바이어스 성분과 랜드 성분을 합산하는 단계; 및

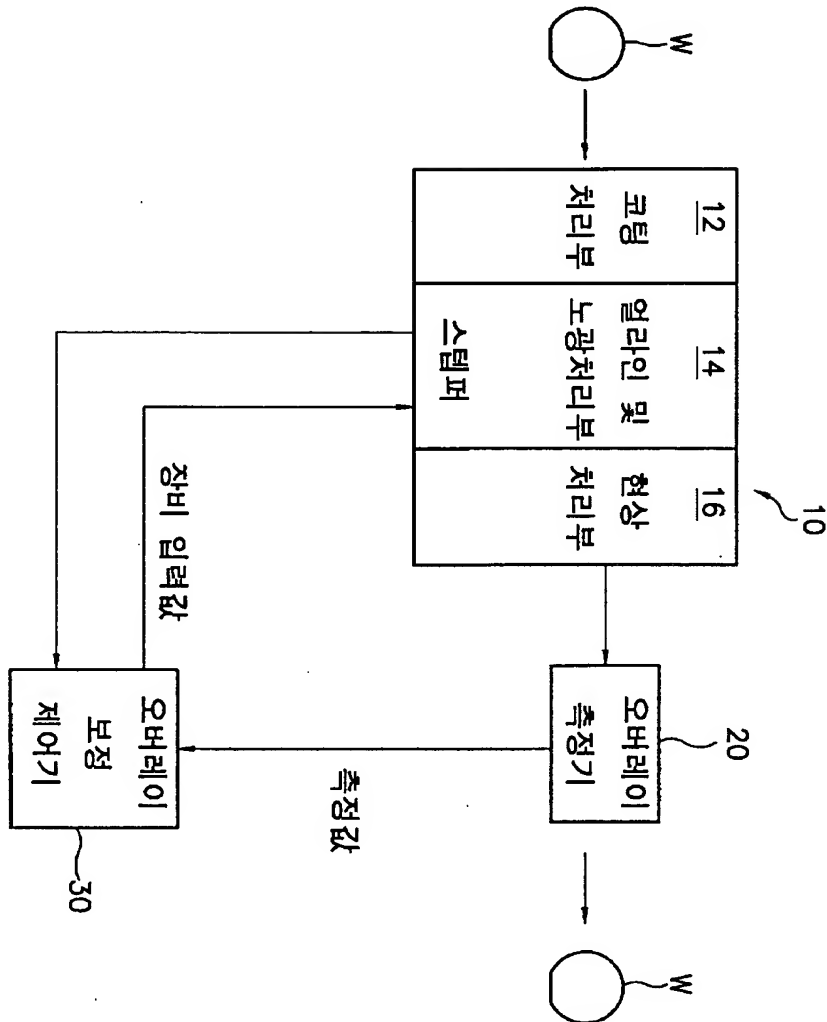
상기 오차값을 사용하여 상기 랜드 성분의 변화를 추종하도록 상기 신경회로망을 학습시키는 단계를 구비하는 것을 특징으로 하는 노광장치의 제어방법.

【청구항 18】

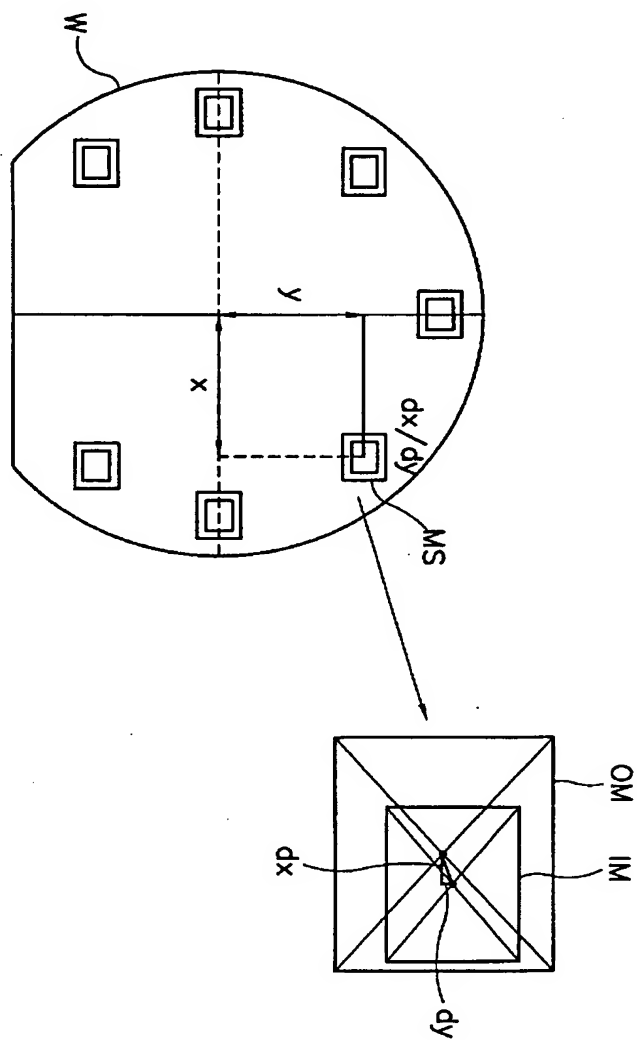
제 17 항에 있어서, 상기 유추단계에서는 각 성분의 보정값에 미치는 영향력이 적은 순서인 BASEII, BASEI, PPID 순으로 유추하는 것을 특징으로 하는 노광장치의 제어방법.

【도면】

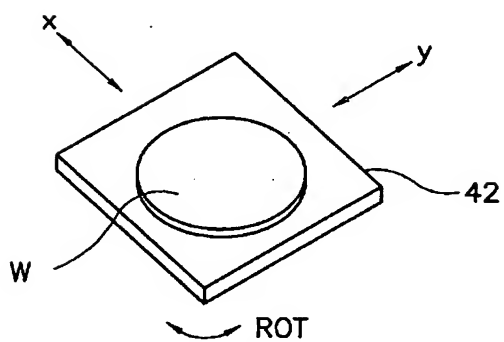
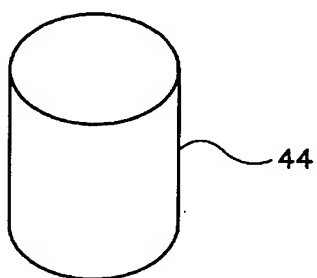
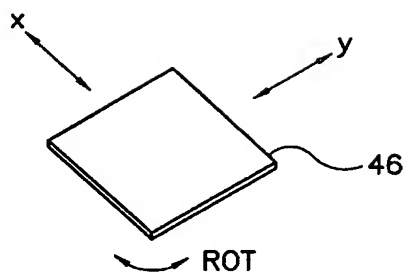
【도 1】



【図 2】



【도 3】

40

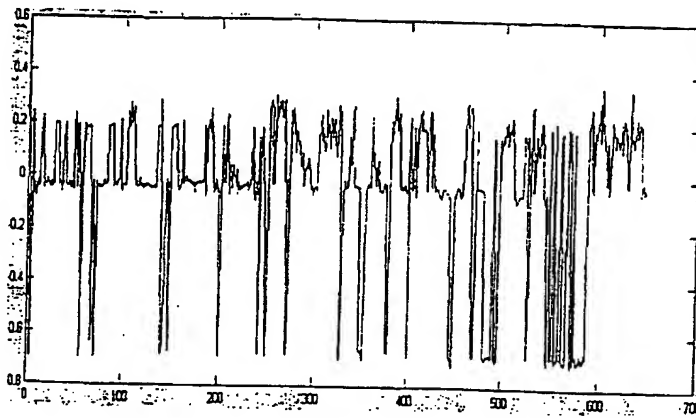
【도 4】

측정 시간
LOT ID
OF-X
OF-Y
SC-X
SC-Y
ORT
W-ROT
RED-X
RED-Y
ROT-X
ROT-Y

【도 5】

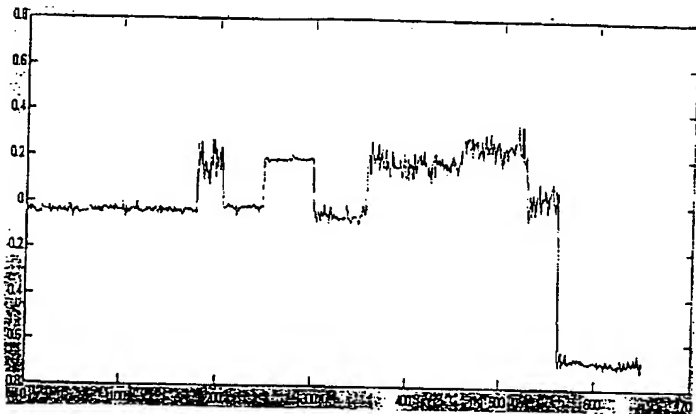
생성 시간			
LOT ID			
FWD OF-X	RET OF-X	NN OF-X	IN OF-X
OF-Y	OF-Y	OF-Y	OF-Y
SC-X	SC-X	SC-X	SC-X
SC-Y	SC-Y	SC-Y	SC-Y
ORT	ORT	ORT	ORT
W-ROT	W-ROT	W-ROT	W-ROT
RED-X	RED	RED	RED
RED-Y			
ROT-X	ROT	ROT	ROT
ROT-Y			

【도 6】

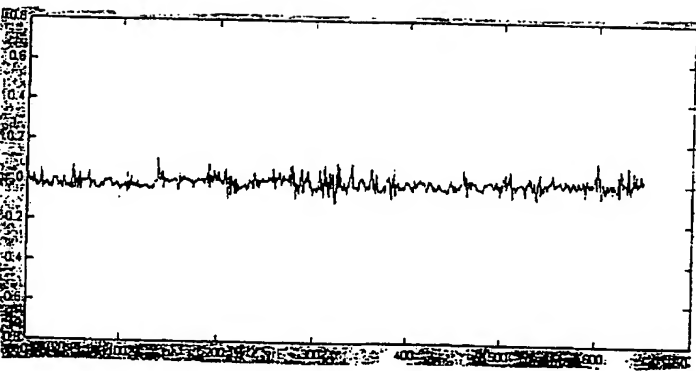


보정 장비 입력 (시간순)

【도 7】

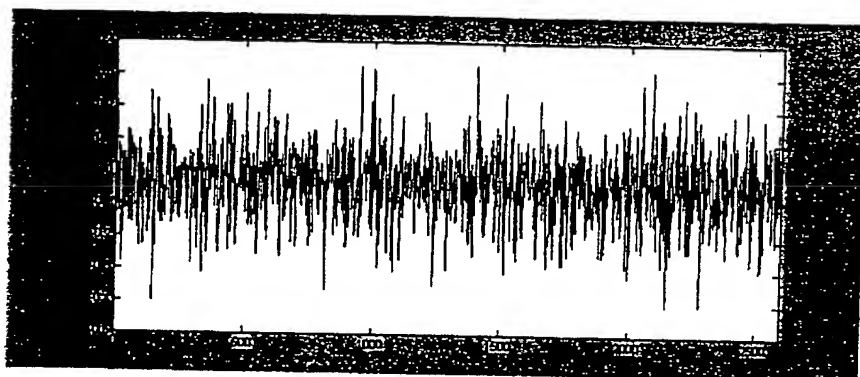


【도 8】

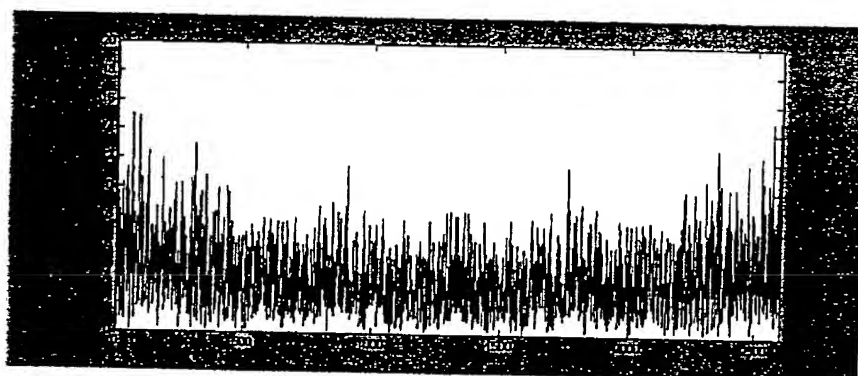


동일 히스토리얼 평균값을 통한 보정 장비 입력

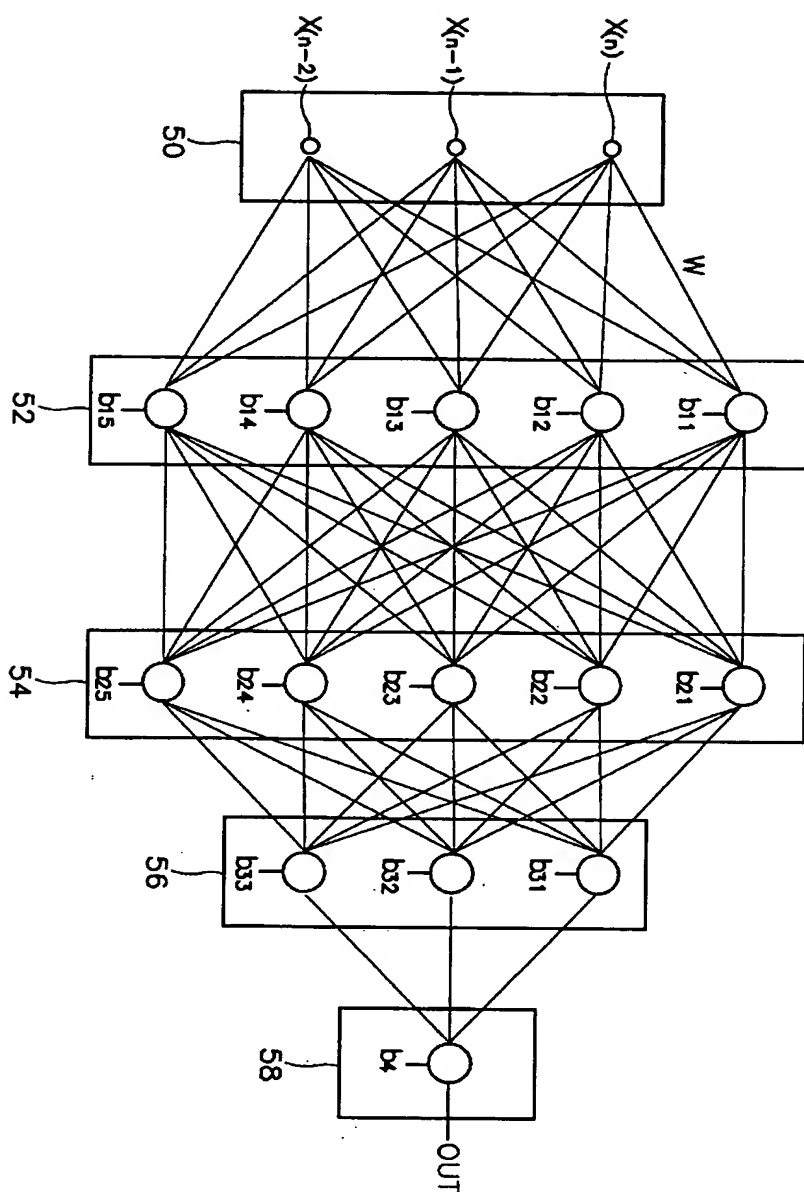
【도 9】

Sample장비1의 offset-x에 대한 $x(n)$

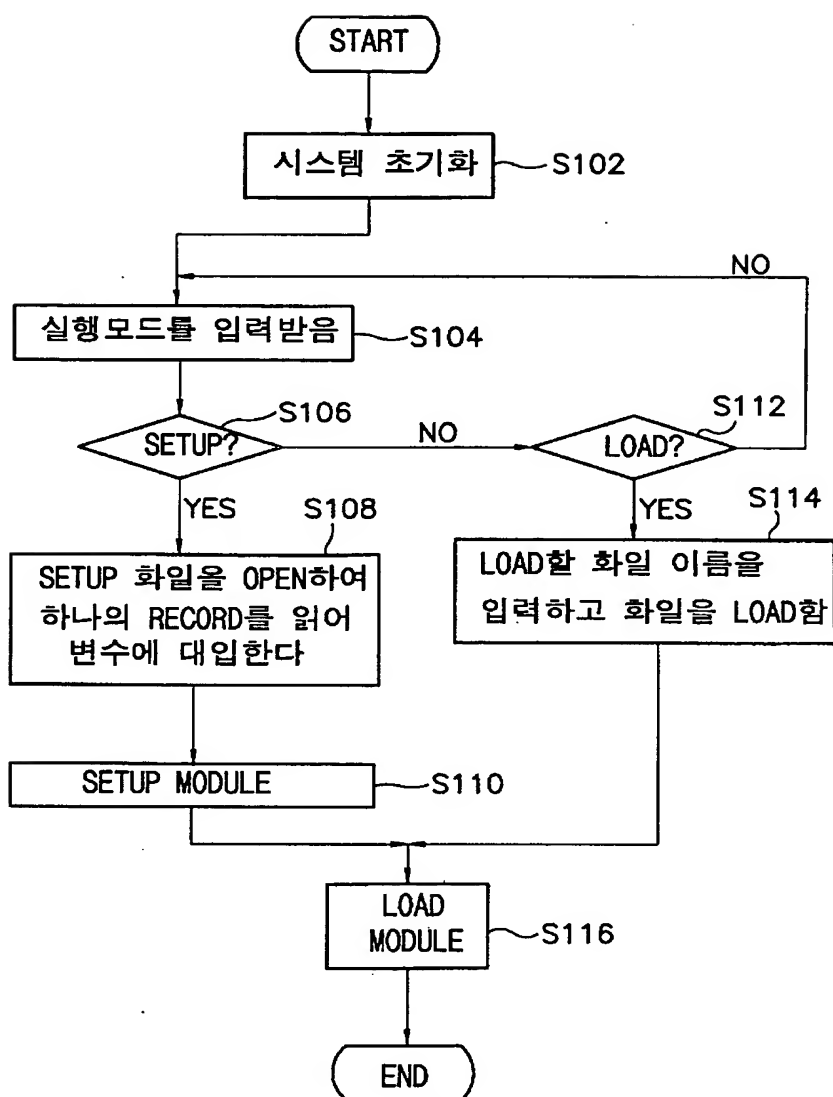
【도 10】

 $x(n)$ 의 주파수 스펙트럼

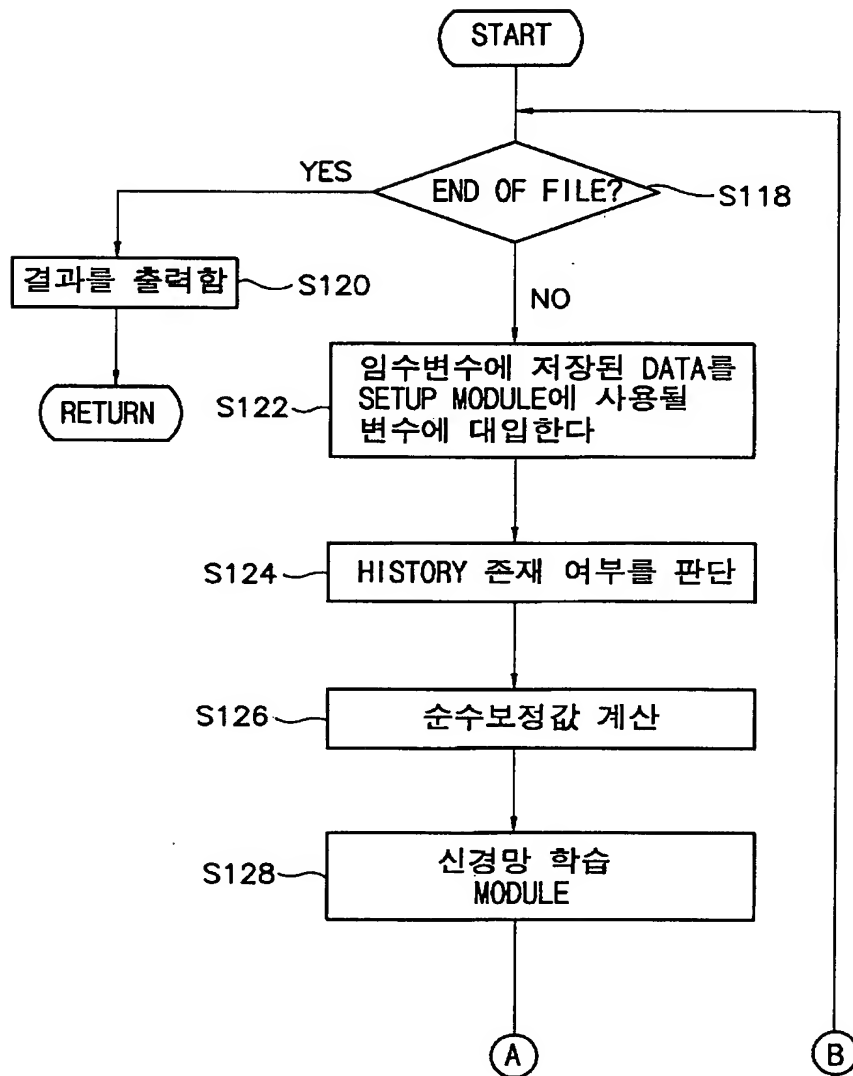
【図 11】



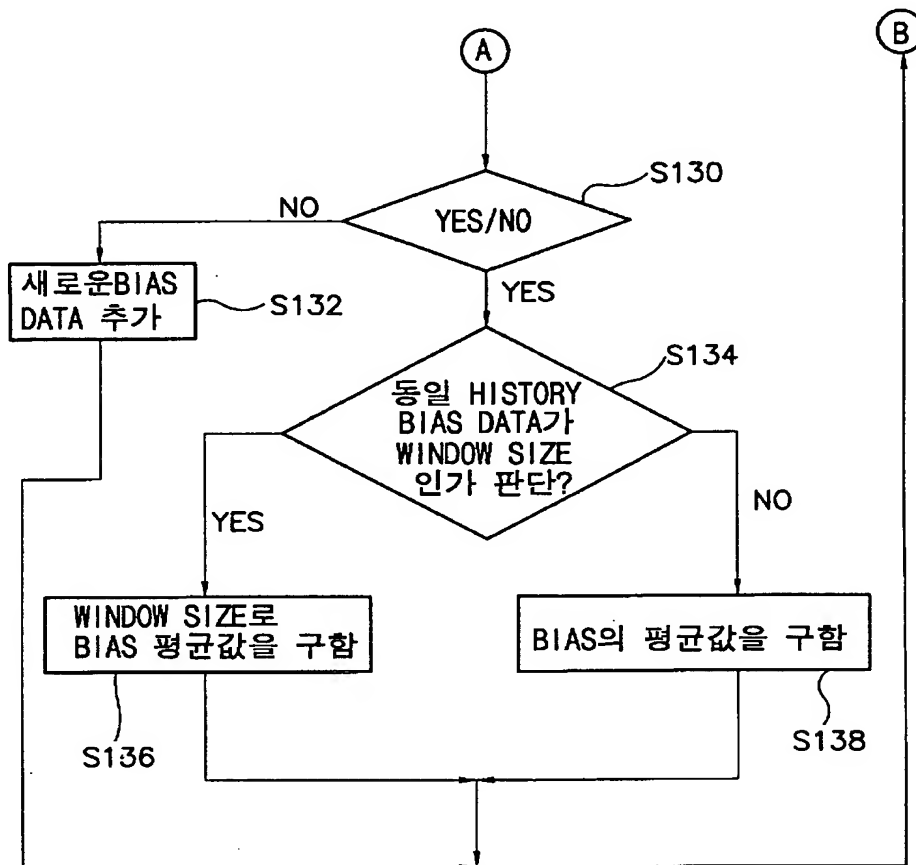
【도 12】



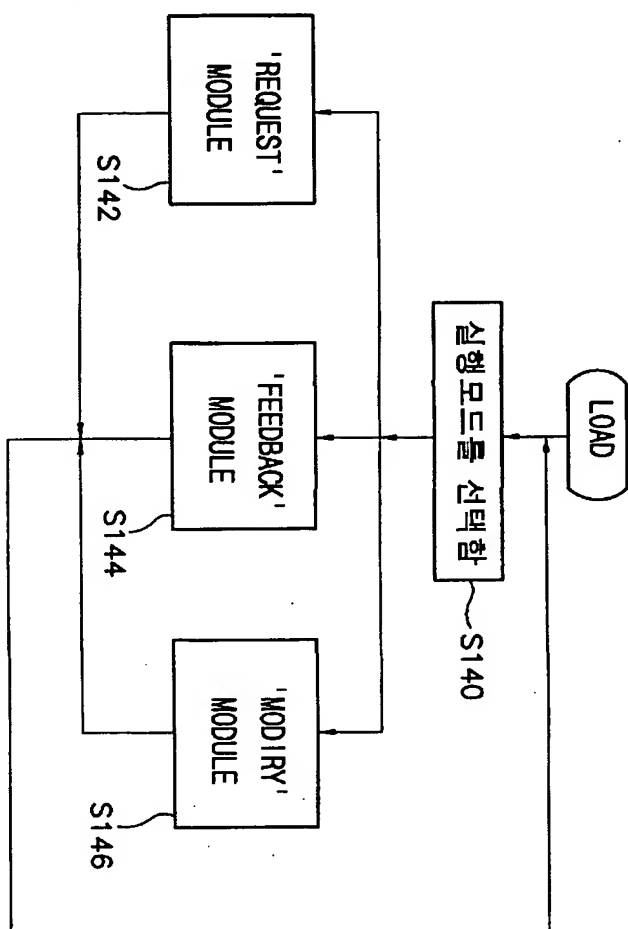
【도 13a】



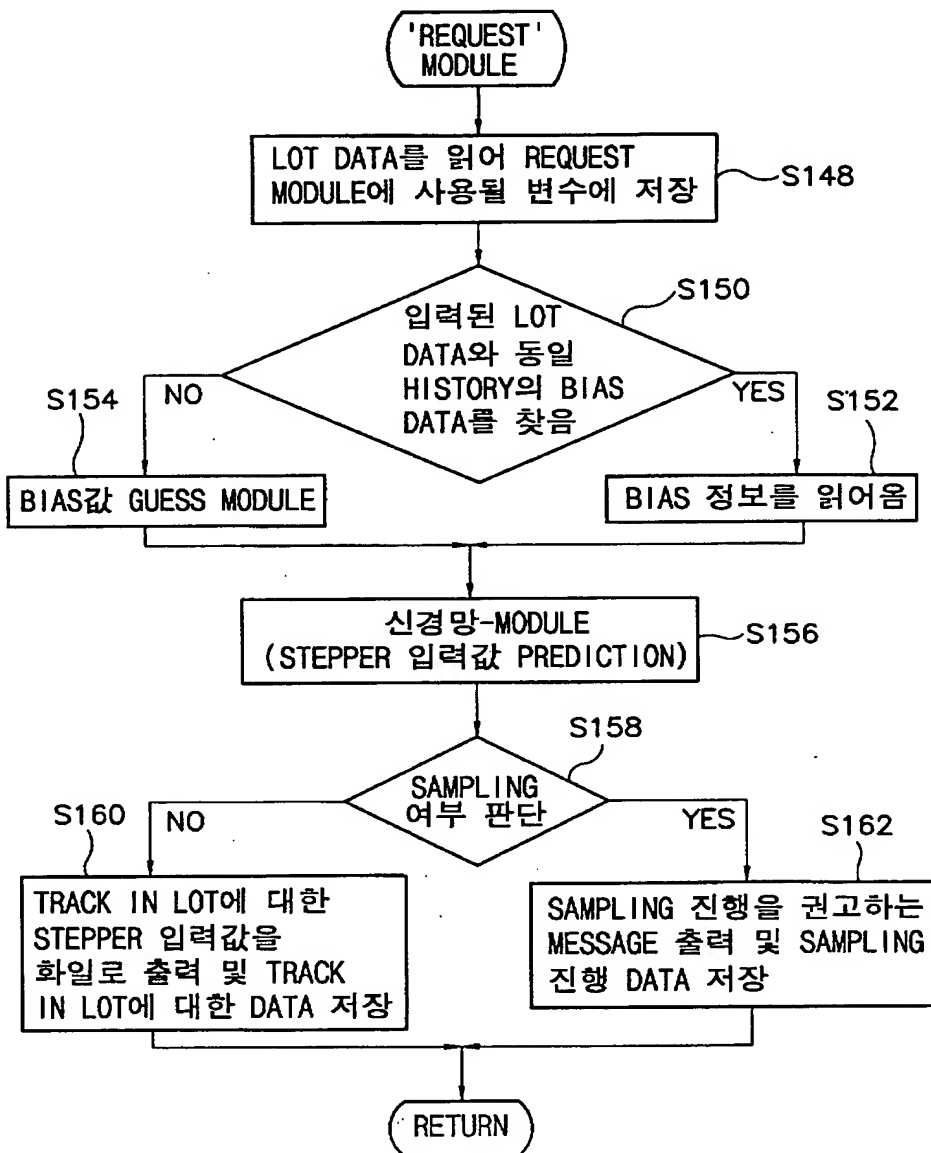
【도 13b】



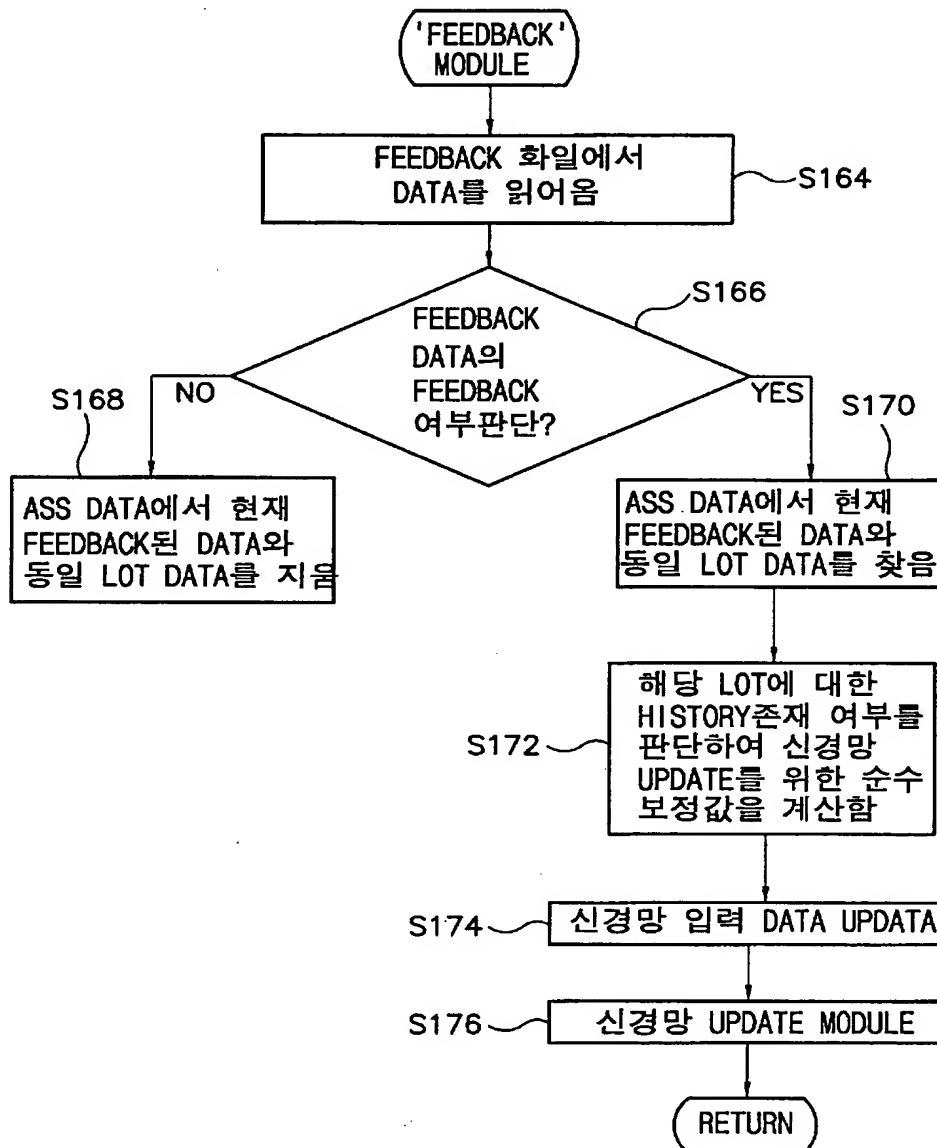
【도 14】



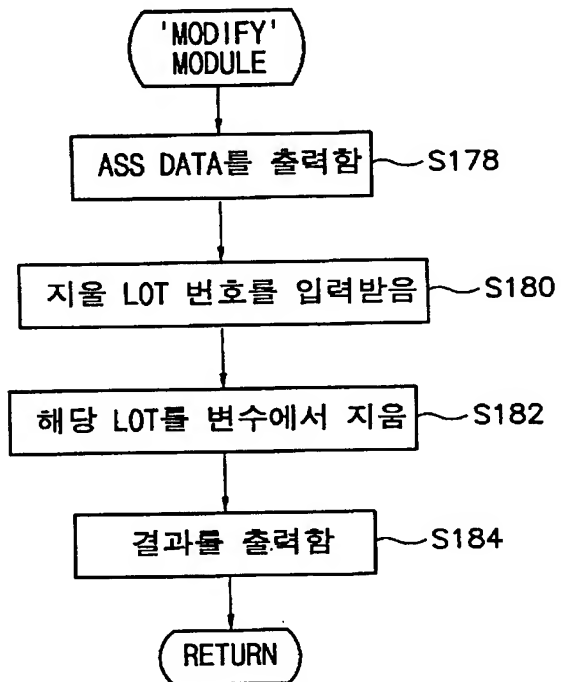
【도 15】



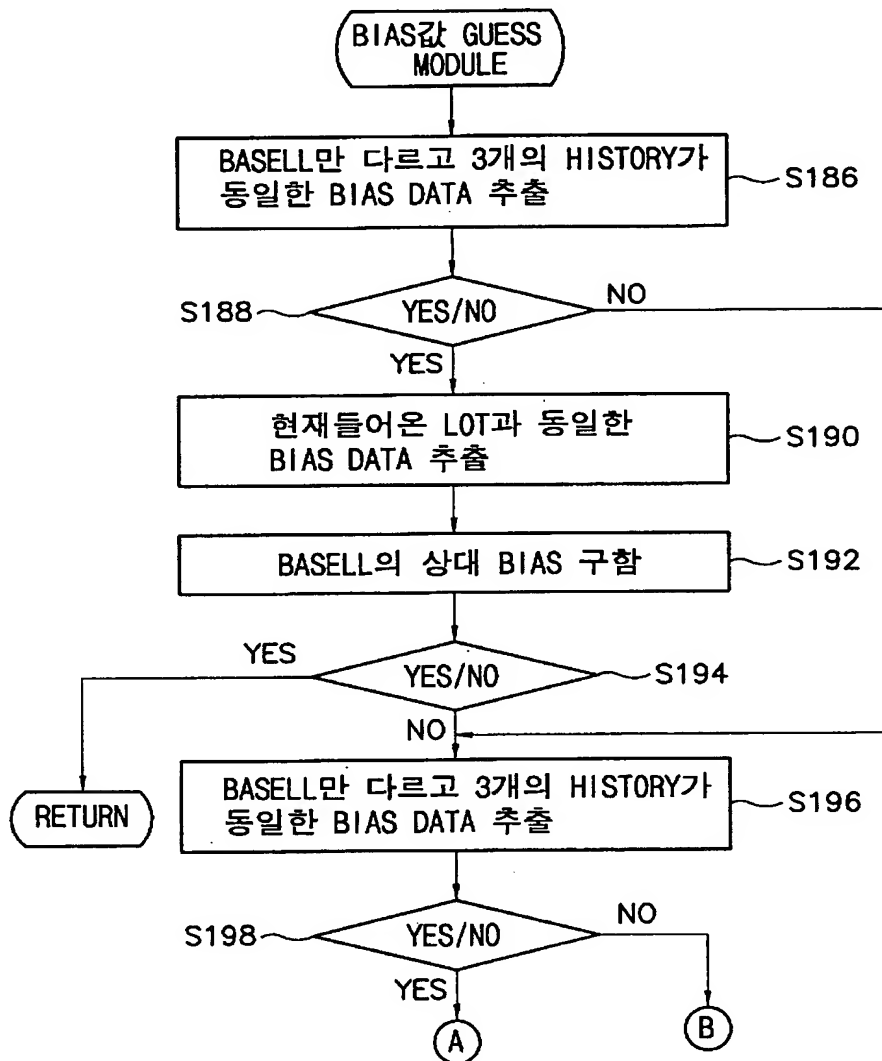
【도 16】



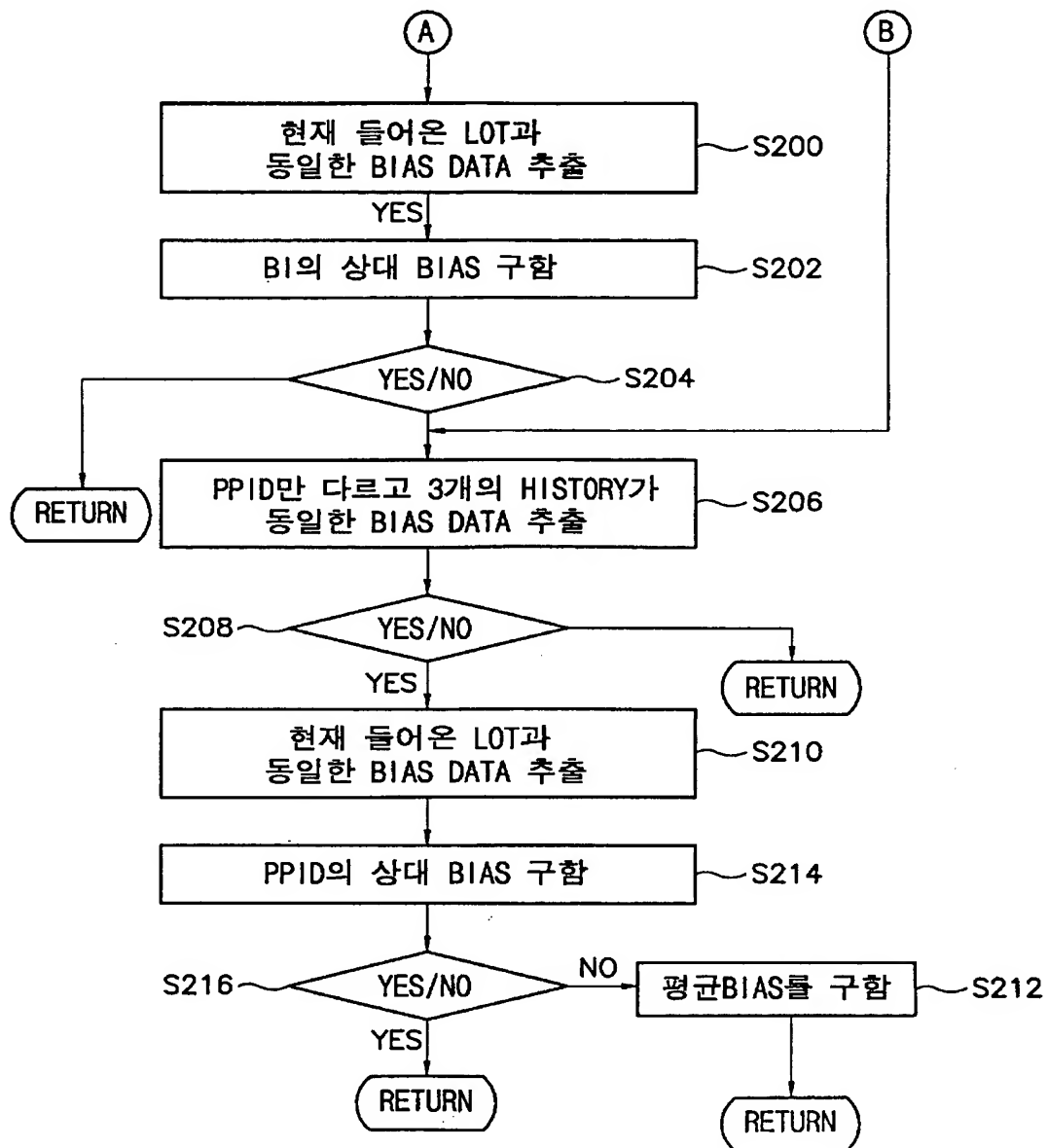
【도 17】



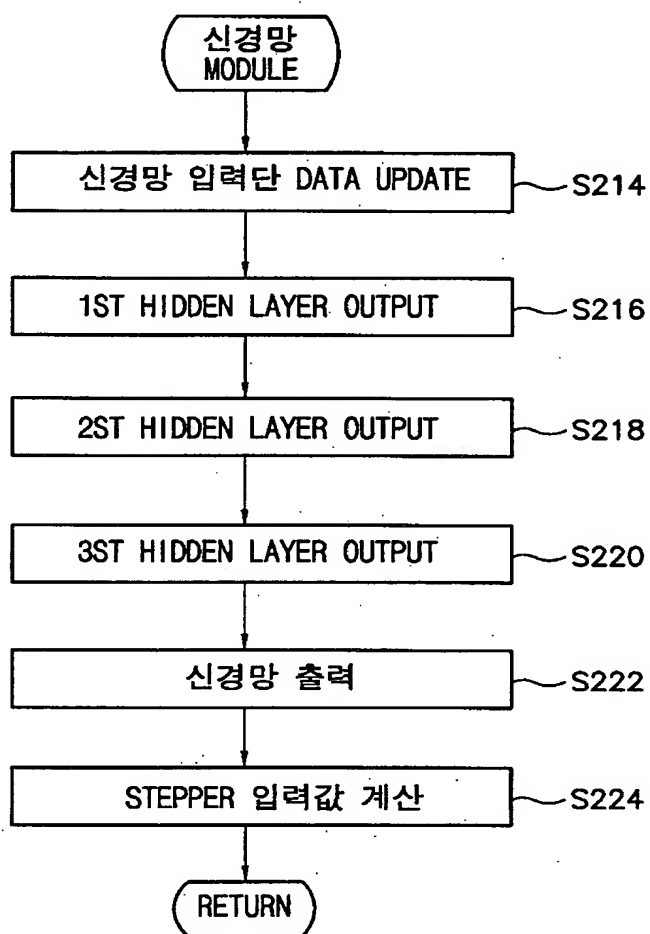
【도 18a】



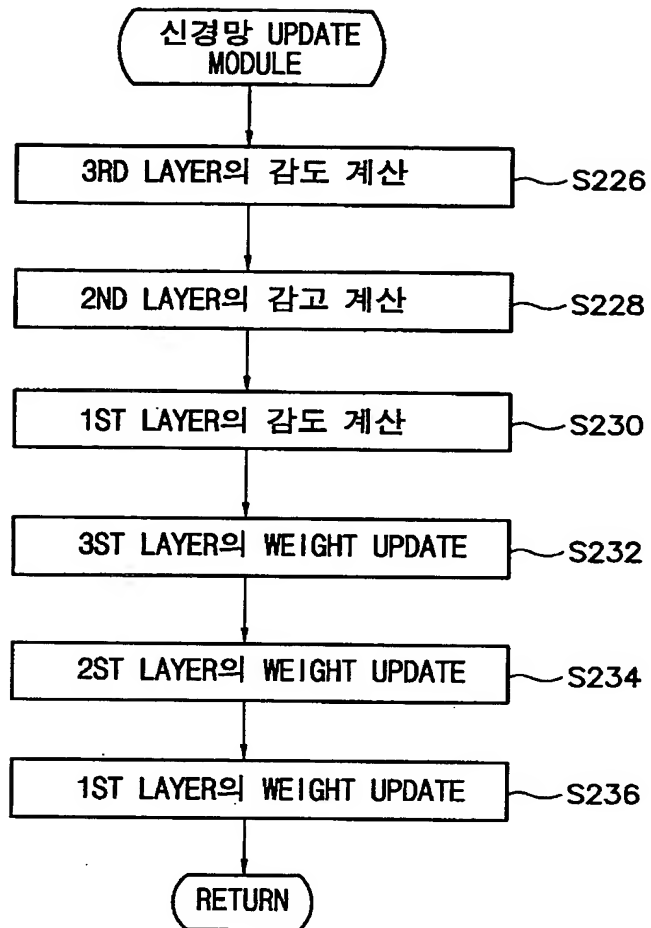
【도 18b】



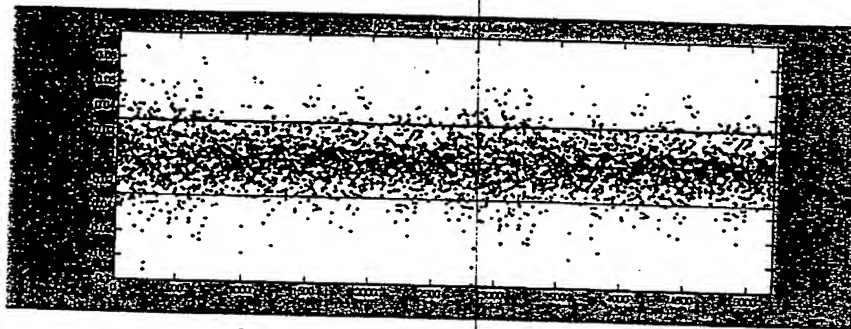
【도 19】



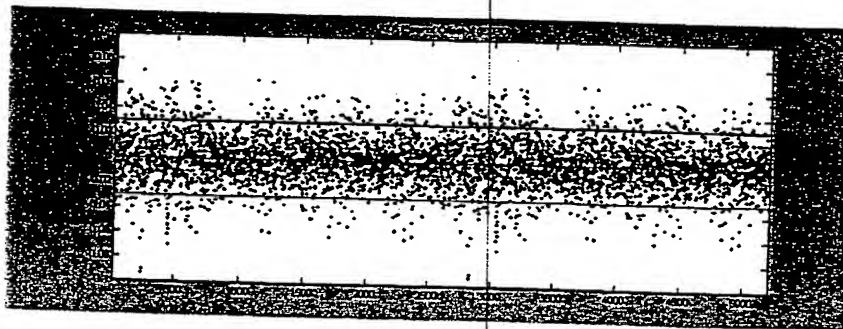
【도 20】



【도 21】

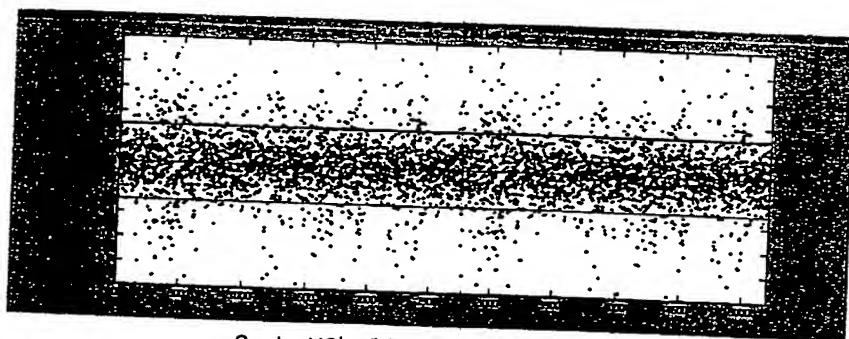


Offset-X의 계측기 측정오차 분포 (ALMS-NN)

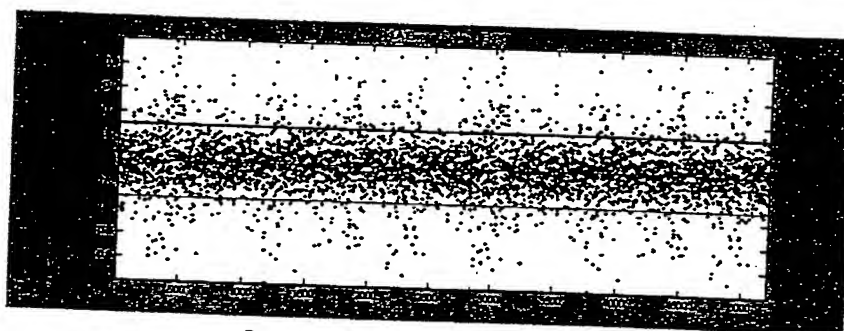


Offset-X의 계측기 측정오차 분포 (기존시스템)

【도 22】

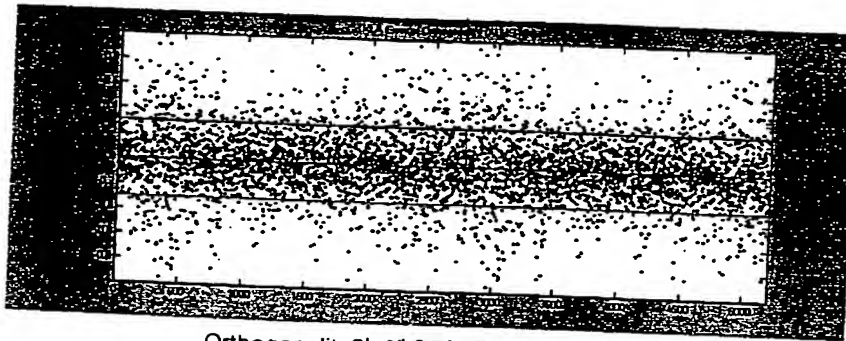


Scale-X의 계측기 측정오차 분포 (ALMS-NN)

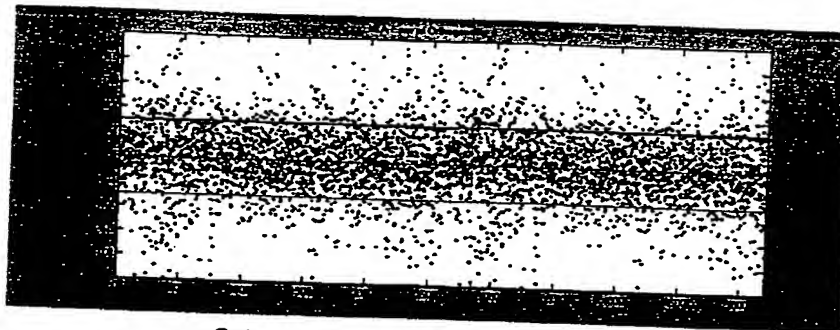


Scale-X의 계측기 측정오차분포(기존시스템)

【도 23】

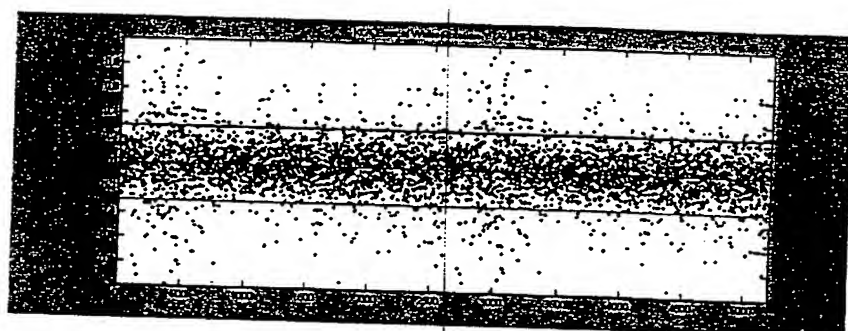


Orthogonality의 계측기 측정오차 분포 (ALMS-NN)

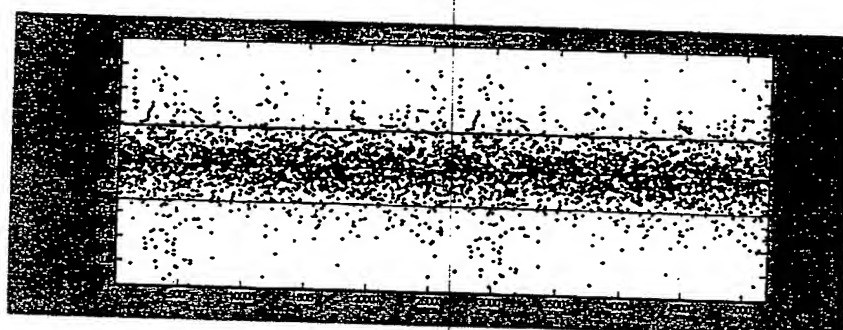


Orthogonality의 계측기 측정오차분포(기존시스템)

【도 24】

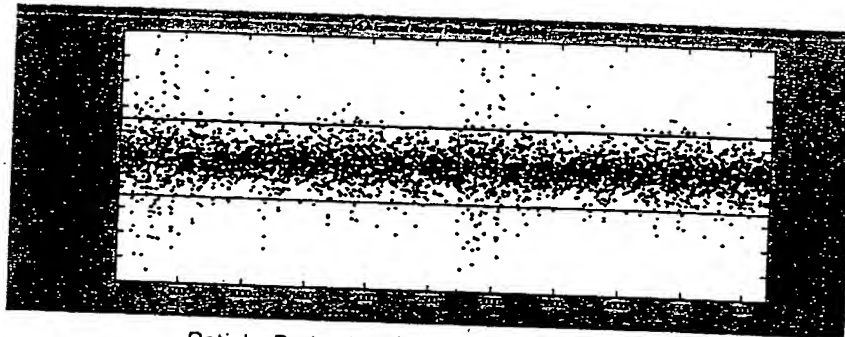


Wafer Rotation-Y의 계측기 측정오차 분포 (ALMS-NN)

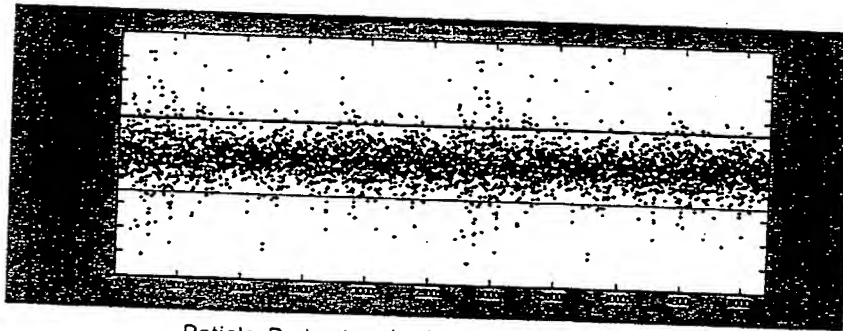


Wafer Rotation-Y의 계측기 측정오차 분포 (기존시스템)

【도 25】

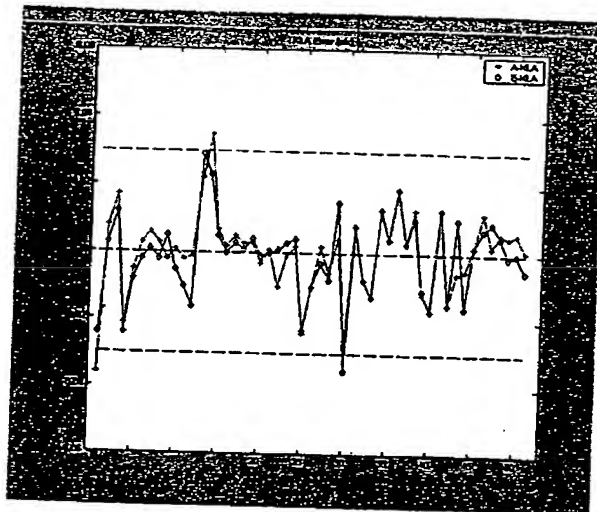


Reticle Reduction의 계측기 측정오차 분포 (ALMS-NN)



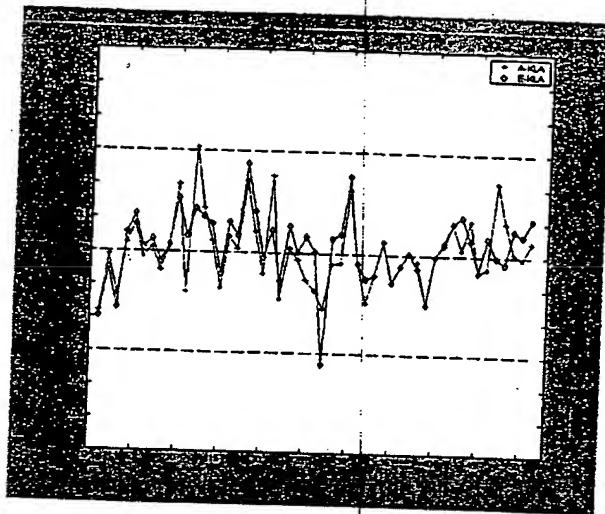
Reticle Reduction의 계측기 측정오차 분포 (기존시스템)

【도 26】



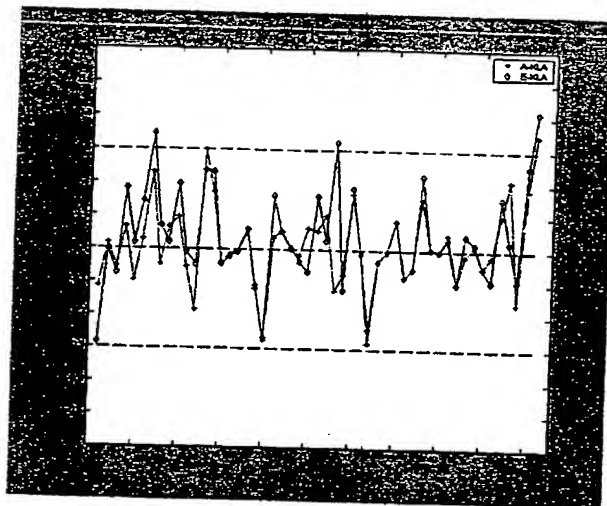
Offset-X의 계측기 측정오차 비교

【도 27】



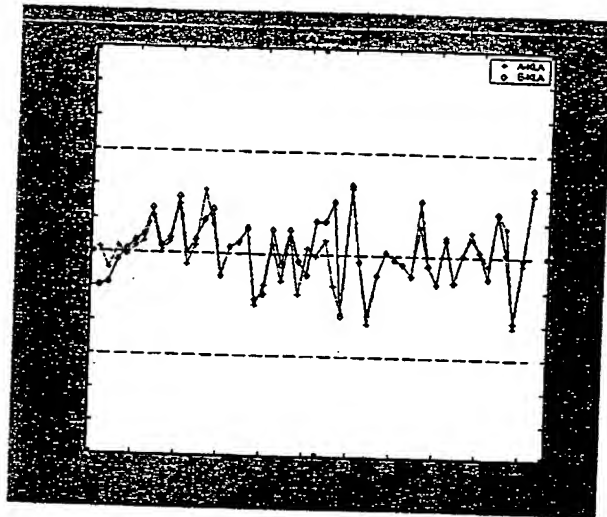
Scale-X의 계측기 측정오차 비교

【도 28】



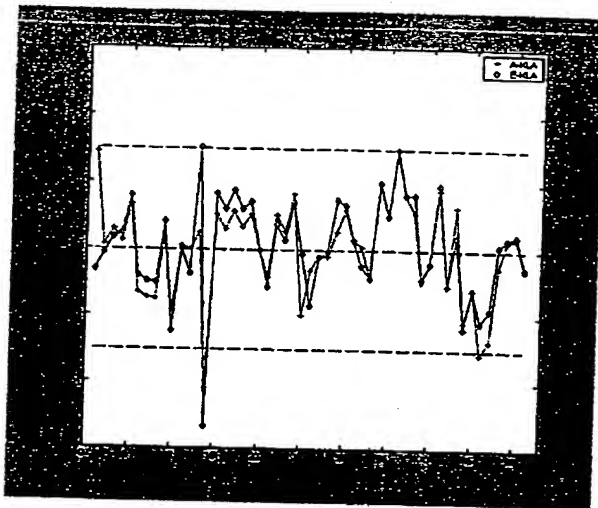
Orthogonality의 계측기 측정오차 비교

【도 29】



Wafer rotation-Y의 계측기 측정오차 비교

【도 30】



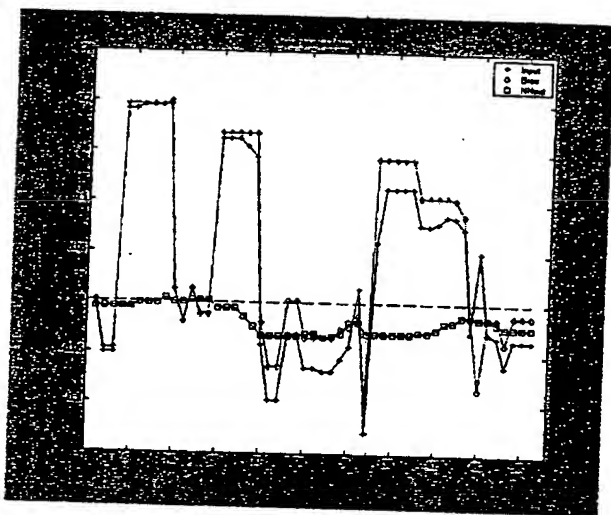
Reticle reduction의 계측기 측정오차 비교

【도 31】



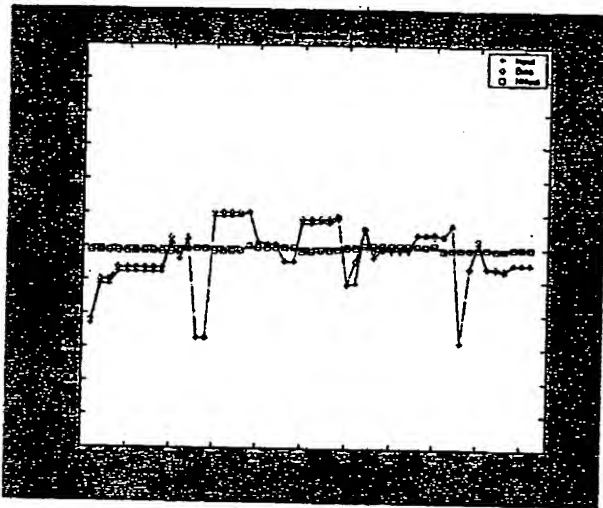
Offset-X의 장비 입력값 구성

【도 32】



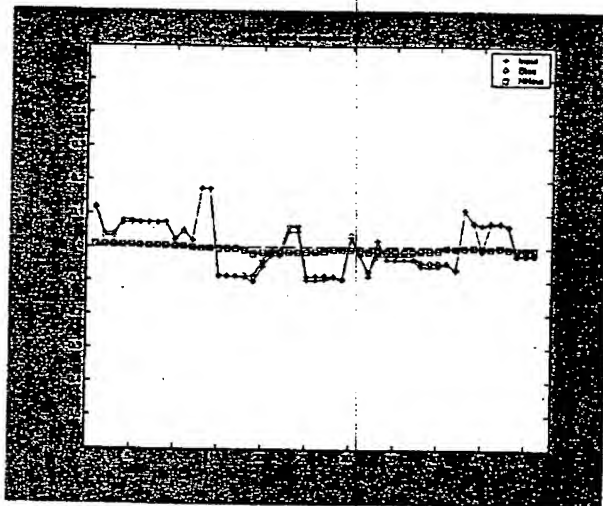
Scale-X의 장비 입력값 구성

【도 33】



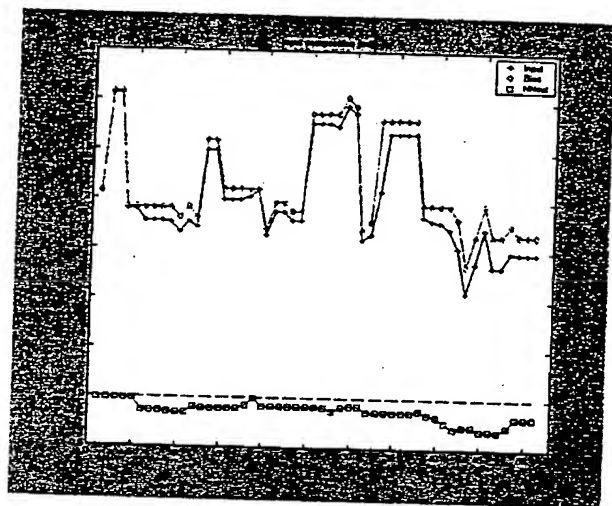
Orthogonality의 장비 입력값 구성

【도 34】



Wafer rotation-Y의 장비 입력값 구성

【도 35】



Reticle reduction의 장비 입력값 구성